

**Pedro Bruno  
Pereira Amaral**

**Aplicação de visualização/manipulação de dados de  
Electrocorticografia**



**Pedro Bruno  
Pereira Amaral**

## **Aplicação de visualização/manipulação de dados de Electrocorticografia**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica de José Maria Fernandes e Paulo Dias, Professores do Departamento de Electrónica, Telecomunicações e Informática, da Universidade de Aveiro



**o júri**

presidente

Professora Doutora Maria Beatriz Alves de Sousa Santos  
Professora Associada com Agregação, Universidade de Aveiro

vogais

Doutor Francisco José Sales Almeida Inácio  
Serviço de Neurologia, Hospitais da Universidade de Coimbra

Professor José Maria Amaral Fernandes  
Professor Auxiliar Convidado, Universidade de Aveiro

Professor Paulo Miguel De Jesus Dias  
Professor Auxiliar Convidado, Universidade de Aveiro



## **agradecimentos**

Gostaria de fazer um agradecimento sincero ao professor José Maria Fernandes e ao professor Paulo Dias por verdadeiramente me **orientarem**, pela ajuda e disponibilidade que me deram ao longo deste ano.

Em meu nome e dos meus orientadores, gostaria de agradecer ao Dr. Francisco Sales, ao neurocirurgião Fernando Gomes e a toda a equipa associada ao programa de cirurgia de epilepsia dos HUC por nos oferecerem tanto o problema como os meios que originaram este trabalho.





## Resumo

Apesar da existência de vários tratamentos para o controlo da epilepsia, quase um terço dos pacientes necessitam de se submeter à cirurgia para controlar as crises. Especialmente nestes casos, é essencial localizar com precisão a área responsável pelas crises.

Com os avanços na tecnologia, o número de técnicas para ajudar no diagnóstico aumentou consideravelmente, assim como a resolução das imagens resultantes. Um cirurgião pode retirar informação sobre a actividade cerebral de diversos exames: electroencefalograma (EEG), ressonância magnética (RM), tomografia computadorizada (TC), tomografia por emissão de positrões (PET) e electrocorticografia (ECoG) onde são usados eléctrodos subdurais ou epidurais para registar o EEG.

Diversas aplicações de software são usadas para analisar estes dados, mas a maior parte delas não suportam funcionalidades para sobrepor na mesma visualização, informação de diferentes modalidades. Na ECoG, por exemplo, é essencial para o cirurgião localizar os eléctrodos no cérebro do paciente para planear a cirurgia e evitar remover áreas críticas.

Nesta tese, é apresentada a primeira versão de um software cujo objectivo é combinar numa mesma aplicação os dados provenientes da ECoG, RMI e TC para fornecer aos cirurgiões uma ferramenta que combine facilmente todos os dados numa mesma aplicação de visualização para facilitar a interpretação da informação disponível como um todo.



## **Abstract**

Despite the existence of several treatments for epilepsy control, nearly one third of the patients still need to undergo surgery in order to control the seizures. Especially in these cases, it is essential to locate precisely the area responsible for the seizures.

With the advances in technology the number of techniques to help in this diagnostic has increased dramatically as well as the resolution of the resulting images. A surgeon can now have information about brain activity from several exams: EEG, MRI, CT, PET and Electrocorticography (ECoG) where epidural or subdural electrodes are used to record the Electroencephalogram (EEG).

Many software applications are used to analyze these data, but most of them do not provide any facility to super-impose in the same view information from different techniques. In ECoG, for example, it is essential for the surgeon to locate the electrodes within the brain of the patient in order to plan the surgery and avoid removing critical areas.

In this thesis, we present the first version of a software whose objective is to combine in the same application the data coming from ECoG, MRI and CT to provide the surgeons with a tool that combine easily all the data in the same visualization tool to facilitate the interpretation of the available information as a whole.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>6</b>
1.1	Epilepsia e o seu tratamento . . . . .	7
1.2	Cirurgia da epilepsia . . . . .	7
1.3	Avaliação Pré-Cirúrgica . . . . .	7
1.3.1	Modalidades não invasivas . . . . .	8
1.3.2	Modalidades invasivas . . . . .	11
1.4	Fusão de Dados . . . . .	11
1.5	Objectivo . . . . .	12
<b>2</b>	<b>Enquadramento</b>	<b>13</b>
2.1	Requisitos do co-registo visual de informação para avaliação invasiva pré-cirúrgica	13
2.1.1	Dados . . . . .	13
2.1.2	Processamento de Dados . . . . .	13
2.1.3	Visualização e Classificação . . . . .	15
2.1.4	Solução e Requisitos . . . . .	15
2.2	Casos de uso . . . . .	16
2.3	Aplicações de Visualização Médica . . . . .	18
2.4	Proposta . . . . .	23
<b>3</b>	<b>BrainGrid - Co-registo Visual de Informação para Avaliação Pré-cirúrgica Invasiva</b>	<b>24</b>
3.1	Organização da Aplicação . . . . .	24
3.2	Escolhas Técnicas . . . . .	25
3.2.1	Visualização . . . . .	25
3.2.2	Processamento de imagem e Co-registo . . . . .	27
3.2.3	Interface Gráfica - GUI . . . . .	28
3.2.4	Formatos Médicos . . . . .	30
3.2.5	Plataforma . . . . .	31
3.3	A Aplicação . . . . .	32
3.3.1	Arquitectura . . . . .	32
3.3.2	Leitura . . . . .	34
3.3.3	Visualização 3D . . . . .	34
3.3.4	Visualização 2D . . . . .	36
3.3.5	Co-Registo . . . . .	36
3.3.6	Segmentação do Cortex . . . . .	38
3.3.7	Extracção de eléctrodos . . . . .	41
3.3.8	Visualização de eléctrodos . . . . .	43

3.3.9	Templates de grelha . . . . .	44
3.3.10	Sincronização . . . . .	45
3.3.11	Visualização de dados eléctricos . . . . .	46
3.3.12	Eléctrodos: propriedades . . . . .	48
3.3.13	Interface com o utilizador . . . . .	49
3.3.14	Inicialização . . . . .	49
3.3.15	Preview . . . . .	51
3.3.16	Co-Reg . . . . .	51
3.3.17	Manual Extractor . . . . .	52
3.3.18	Extractor . . . . .	53
3.3.19	BrainGrid . . . . .	54
<b>4</b>	<b>Resultados</b>	<b>57</b>
4.1	Extracção de eléctrodos . . . . .	58
4.1.1	Automática . . . . .	58
4.1.2	Manual . . . . .	59
4.2	BrainGrid . . . . .	61
<b>5</b>	<b>Conclusão</b>	<b>65</b>
	<b>Referências</b>	<b>66</b>
<b>A</b>	<b>- Arquitectura</b>	<b>70</b>
<b>B</b>	<b>- Templates de grelhas de eléctrodos</b>	<b>78</b>
<b>C</b>	<b>- Outros resultados</b>	<b>82</b>
<b>D</b>	<b>- Instalação</b>	<b>89</b>

# Lista de Figuras

1.1	Exemplo de aquisição de EEG . . . . .	8
1.2	Exemplo RMI . . . . .	9
1.3	Exemplo TC . . . . .	9
1.4	Colocação de uma grelha para ECoG . . . . .	12
2.1	Diagrama Conceptual . . . . .	14
2.2	Visão funcional sobre o sistema . . . . .	16
2.3	MRlcro . . . . .	19
2.4	VolView . . . . .	19
2.5	3D Slicer . . . . .	20
2.6	EtDips . . . . .	20
2.7	ezDICOM . . . . .	21
2.8	Julius . . . . .	21
2.9	AMIDE . . . . .	22
2.10	BrainSuite 2 . . . . .	22
2.11	Proposta de Interface . . . . .	23
3.1	Diagrama da organização da aplicação . . . . .	25
3.2	Exemplo VTK - Isocontouring com cortes . . . . .	26
3.3	Pipeline de visualização do VTK . . . . .	26
3.4	Algumas <i>widgets</i> do KWWidgets . . . . .	29
3.5	Arquitectura . . . . .	32
3.6	Vista Geral do diagrama de classes . . . . .	33
3.7	Pipeline de Visualização 3D . . . . .	35
3.8	Testes feitos com a tecnica Volume Ray Cast, variando a função transferencia de cor, opacidade, e propriedades de luz do volume . . . . .	36
3.9	Pipeline de Visualização 2D . . . . .	37
3.10	Componentes básicos da framework de co-registo: as duas imagens, a transformação, a métrica, o interpolador e o optimizador ( <i>Fonte: The ITK Software Guide</i> ) . . . . .	37
3.11	Comparação entre BET, AFNI e BSE. . . . .	38
3.12	BET <i>flowchart</i> [Smith, 2002] ( <i>Fonte: FMRIB technical Report TR00SMS2b</i> ) . . . . .	40
3.13	Algoritmo de extracção de eléctrodos . . . . .	41
3.14	Vistas 2D e 3D do interface de extracção manual. . . . .	42
3.15	Lista de pontos no interface de extracção manual. . . . .	43
3.16	Pipeline de colocação de eléctrodos . . . . .	43

3.17	Exemplo de template de grelha (fictícia)	44
3.18	Vista 3D e vistas 2D sincronizadas.	45
3.19	Exemplo de uma triangulação de Delaunay para 100 pontos	46
3.20	Pipeline para criação da superfície da grelha de eléctrodos	47
3.21	Visualização de dados de ECoG	47
3.22	Interface de classificação de eléctrodos e sua visualização no template da grelha	48
3.23	Interface Inicial	49
3.24	Janela do preview.	51
3.25	Janela do Co-Reg e algoritmo de co-registo a correr.	51
3.26	Interface de Extracção Manual de Eléctrodos	52
3.27	Interface de Extracção de Eléctrodos	53
3.28	Interface Principal	54
3.29	Side Bar - todas as opções não mostradas no interface principal	56
4.1	Extracção de eléctrodos - selecção da imagem	58
4.2	Extracção de eléctrodos - sequência da selecção de eléctrodos.	58
4.3	Extracção manual de eléctrodos - parte 1	59
4.4	Extracção manual de eléctrodos - parte 2 e 3	59
4.5	Vista 3D, alteração de opacidades para melhor visualização dos eléctrodos	60
4.6	Sequência de selecção dos dados a visualizar	61
4.7	Visualização 3D e 2D com zoom 2x.	61
4.8	Inclusão da grelha de eléctrodos e alinhamento de vistas 2D com uso da mira e visualização da ECoG.	62
4.9	Alteração dos meta-dados e gravação do estado da marcação	63
4.10	Exemplo	64
A.1	Pacote Visualização	70
A.2	Pacote Dados	71
A.3	Pacote Load/Save	71
A.4	Pacote ECoG	72
A.5	Pacote Inicialização	73
A.6	Pacote Tratamento de Dados	73
A.7	Pacote GUI	74
A.8	Pacote Auxiliar	74
A.9	Pacote Navegação de Meta-Dados	75
A.10	Pacote Sliders	76
A.11	Pacote Sincronização	76
A.12	Pacote Pickers	77
B.1	Grelha rectangular 4x8 - 32 eléctrodos	78
B.2	Grelha rectangular 2x6 - 18 eléctrodos	79
B.3	Grelha em fita 1x6 - 6 eléctrodos	80
B.4	Grelha em leque - 24 eléctrodos	81
C.1	Caso 1	83
C.2	Caso 2	84
C.3	Caso 3	85
C.4	Caso 4	86

---

C.5	Caso 5 . . . . .	87
C.6	Caso 6 . . . . .	88



# Capítulo 1

## Introdução

A epilepsia é uma condição crónica caracterizada por crises recorrentes e espontâneas [Gastaut, 1973], que afecta cerca de 1 em cada 200 indivíduos [Hauser, 1998], o que pressupõe, para Portugal, uma prevalência entre 30 a 50 mil pessoas com epilepsia. Existem uma variedade de causas, que incluem tumores, malformações congénitas, lesões perinatais, alterações genéticas, traumatismos, doenças vasculares e infecciosas. O termo epilepsia abarca em si, várias síndromes associadas com diversos tipos de crises não provocadas. Estas crises são fenómenos electro-fisiológicos anormais temporários, que resultam numa sincronização anormal da actividade eléctrica neuronal.

Podem manifestar-se como uma alteração do estado mental, movimentos descontrolados, convulsões e outros fenómenos psíquicos. Apenas se considera como epilepsia o aparecimento de duas ou mais crises de início brusco e inesperado. Uma crise isolada não é epilepsia; assim, as crises surgidas no decurso de um processo agudo: febre, infecção, ou outros, denominam-se de crises ocasionais e não constituem uma epilepsia [Crawford, 2000]. Na maioria dos casos, os pacientes sofrem de crises localizadas ou focais. Estas crises focais começam com a activação de um pequeno número de neurónios, habitualmente num dos hemisférios cerebrais, as quais podem manter-se localizadas ou estender-se secundariamente (generalizar-se) através de todo o cérebro.

Torna-se assim evidente a necessidade do estudo desta doença e o desenvolvimento de tratamentos para ela.

## 1.1 Epilepsia e o seu tratamento

O objectivo do diagnóstico clínico da epilepsia é identificar, com a maior precisão possível, a zona do cérebro responsável pelas descargas eléctricas anormais, causadoras da crise - zona epileptogénica (ZE) [Rosenow, 2001] - de forma a melhor controlar as suas ocorrências e melhorar a qualidade de vida dos pacientes epilépticos. Este controlo pode ser feito por duas vias: por meio de medicamentos ou por meios cirúrgicos. Apesar dos avanços no tratamento da epilepsia através de medicação (AED - *Anti-Epileptic Drugs*) nas últimas décadas, aproximadamente um terço dos pacientes com epilepsia continuam resistentes a estes tratamentos [Crawford, 2000]. Este facto traduz-se num agravamento ao longo do tempo da sua condição médica, física e social, apesar das várias estratégias usadas, nomeadamente terapêuticas envolvendo múltiplos medicamentos. Neste contexto, o tratamento cirúrgico pode ser uma opção.

## 1.2 Cirurgia da epilepsia

De um ponto de vista terapêutico, a cirurgia da epilepsia é um tratamento seguro e eficaz em particular em pacientes com epilepsia focal resistente, como demonstrado por uma série de testes controlados em que se compararam os resultados da cirurgia versus terapêuticas baseadas em drogas anti-epilepticas (AED) [Wiebe, 2001].

O objectivo da cirurgia da epilepsia é a remoção da zona epileptogénica (ZE), zona responsável pelas crises [Engel, 1993]. A prática de remoção cirúrgica das áreas cerebrais responsáveis pelas crises iniciou-se há cerca de 50 anos. Contudo, o tratamento cirúrgico é normalmente reservado apenas aos casos em que a medicação não surte efeito, e que com as novas técnicas cirúrgicas e métodos para identificar a zona epileptogénica, oferecem cada vez mais, um melhor prognóstico [Wiebe, 2001].

Em combinação com as técnicas de diagnóstico mais modernas, a cirurgia da epilepsia traz vantagens tanto a nível de custos em relação a uma epilepsia não controlada como em relação à redução dos riscos inerentes à mesma.

## 1.3 Avaliação Pré-Cirúrgica

No entanto uma avaliação pré-cirúrgica é essencial para aumentar as probabilidades de sucesso da cirurgia através da delineação da zona epileptogénica.

A avaliação pré-cirúrgica tem como objectivo localizar a zona epileptogénica e planear a cirurgia de forma a assegurar a sua extracção e garantir que a cirurgia não vai lesar nenhuma função cerebral importante como por exemplo as zonas onde residem a linguagem ou a memória [Luders, 1992, Rosenow, 2001].

O processo de avaliação pré-cirúrgica é suportado em várias modalidades para diagnóstico, umas invasivas - envolvem um procedimento cirúrgico - e não invasivas [Duncan, 2002, Richardson, 2003, Knowlton, 2004, Koepp, 2005].

### 1.3.1 Modalidades não invasivas

As modalidades não invasivas usadas mais frequentemente na avaliação pré-cirúrgica são:

- Electroencefalograma (EEG)
- Ressonância Magnética Nuclear (RMN)
- Tomografia Computadorizada (TC)
- Tomografia de emissão de positrões (PET)
- Tomografia computadorizada de emissão de fotões (SPECT)
- Registo video-EEG simultâneo
- Testes Neuropsicológicos

O **Electroencefalograma (EEG)** (ver Figura 1.1) é uma medição neurofisiológica da actividade eléctrica do cérebro através de eléctrodos colocados no escalpe [Malmivuo, 1997]. O sinal eléctrico medido é resultado de actividade eléctrica de áreas corticais que podem envolver grande número de neurónios. Através das variações desta actividade eléctrica cerebral é possível associar padrões de actividade eléctrica medidos no escalpe com áreas cerebrais, nomeadamente onde as crises ocorrem.



Figura 1.1: Exemplo de aquisição de EEG

A **Ressonância Magnética Nuclear (RMN)** ou apenas Ressonância Magnética (RM) (ver Figura 1.2) é usada principalmente em imagiologia médica para mostrar modificações patológicas ou fisiológicas de tecidos vivos [Grant, 2004]. É tipicamente usada para caracterizar espacialmente a morfologia do cérebro, nomeadamente na identificação de malformações e lesões (mormente cicatrizes ou lesões resultantes de um deficiente desenvolvimento cerebral). Dada a sua resolução espacial, as RM do cérebro são usualmente usadas como referência para o mapeamento e co-registo espacial de outras modalidades com a estrutura cerebral.

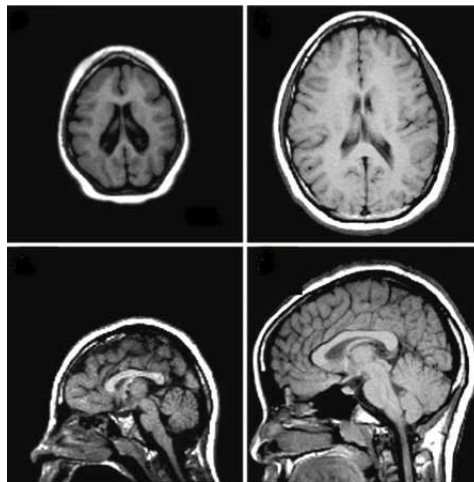


Figura 1.2: Exemplo RMI

A **Tomografia Computadorizada (TC)** (ver Figura 1.3), originalmente apelidada de Tomografia Axial Computadorizada (TAC), é obtida através do processamento por computador de informação recolhida após expor o corpo, ou zona do mesmo, a uma sucessão de raios X. É especialmente exacta na caracterização espacial de tecidos ósseos.

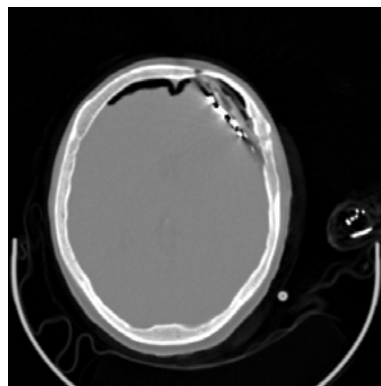


Figura 1.3: Exemplo TC

A **Tomografia de emissão de positrões (PET)** através de marcadores químicos marcados radiologicamente, permite, em certos casos, identificar áreas associadas a certos eventos metabólicos [Koepp, 2005, Duncan, 1997]. No caso particular da epilepsia, com ajuda de marcadores de glucose entre outros, pode ajudar a identificar áreas com metabolismo anormal que muitas vezes estão associadas à área epileptogénica.

A **Tomografia computadorizada de emissão de fótons (SPECT)** é uma técnica de imagiologia médica nuclear que usa raios gamma que através de um marcador radioactivo permite identificar áreas onde a absorção de oxigénio é mais elevada/mais reduzida em relação aos padrões normais [Duncan, 1997]. Através de um processo de amostragem em todo o cérebro seguido de um processo de reconstrução consegue produzir verdadeira informação 3D. No caso da epilepsia padrões anormais de metabolismo podem ser associados áreas responsáveis pelas crises. Tipicamente usado como pista para identificar a ZE, assumindo normalmente que nesta, durante as crises, é observada uma maior actividade.

O **registo video-EEG simultâneo** consiste na aquisição de sinal EEG acompanhadas de gravação vídeo em simultâneo. Tem como propósito relacionar os padrões epileptiformes do EEG (e.g. crises) com as manifestações visíveis (semiologia), especialmente durante as crises (e.g. movimentos involuntários, etc ) [Cascino, 2002]. Da relação entre o EEG e a semiologia das crises é possível inferir o local do início da crise ou distinguir crises epilépticas de crises não epilépticas. Os registos vídeo-EEG envolvem normalmente a hospitalização do paciente por um período de tempo, tipicamente dias a semanas, durante o qual são monitorizados de forma a aumentar as probabilidades de se observarem crises.

Os **testes Neuropsicológicos** cobrem áreas tão diversas como a memória, a linguagem, a personalidade e o pensamento. Podem ajudar a localizar zonas de défice funcional, muitas vezes relacionadas com a localização da origem da crise. Fornecem informação de base e são normalmente repetidos após a cirurgia para verificar alterações no funcionamento da mente.

O processo de cirurgia para epilepsia pode ser longo e difícil. A avaliação psiquiátrica pode ajudar o paciente a apontar para objectivos e expectativas razoáveis, assim como prepara-lo para a cirurgia e fase de recuperação.

### 1.3.2 Modalidades invasivas

Por vezes as modalidades não invasivas não são suficientes para localizar com precisão a zona epileptogénica e zonas funcionais relevantes. Modalidade invasivas como a **Electrocorticografia** (ECoG) [Enger and Burr, 1994] [Jayakar, 1999] são uma opção que permite registar a actividade eléctrica na superfície do córtex cerebral ao invés do EEG que adquire no escalpe. A ECoG implica o internamento para acompanhamento clínico. O internamento aumenta as probabilidades de observação de crises e em simultâneo, o mapeamento de funções cerebrais através da estimulação cortical. Esta permite relacionar eléctrodos específicos com alterações observadas durante a estimulação através do EEG, ECoG e vídeo.

## 1.4 Fusão de Dados

Neste contexto, conjugar todas a informação disponível para suportar a decisão e planeamento pré-cirúrgico é uma tarefa essencial que se pode revelar tecnicamente difícil pois cada uma das modalidades emprega uma técnica diferente para adquirir os dados e cada imagem resultante tem diferentes características nem sempre directamente comparáveis.

Um bom exemplo é a sobreposição de imagens de RMI e SPECT relativas ao mesmo sujeito (SISCOM) em casos mais complexos de diagnósticos de epilepsia [O'Brien, 1999]. Numa perspectiva mais técnica, o co-registo pode ser definido como o processo de mapear duas imagens da mesma região do corpo com precisão para que cada ponto numa imagem esteja mapeado no correspondente da segunda imagem, permitindo assim uma visualização combinada e geometricamente rigorosa.

A grande dificuldade do co-registo de imagens é determinar com precisão esse mapeamento. O co-registo pode ser manual através da manipulação das imagens dispostas no monitor com auxílio de um rato ou *trackball*. Contudo este método pode não ser o mais preciso, pois muitas vezes não existe informação suficiente, nomeadamente pontos de referência comuns às várias modalidades.

A utilização de marcadores pode ser uma solução para este problema. Usando marcadores em localizações pré-definidas no paciente durante realização dos exames que sejam visíveis nas várias modalidades para possibilitar os seus posterior alinhamento.

No entanto este processo pode ser dificultado, por exemplo, por movimentos do paciente ou outros artefactos que possam ter efeitos ou induzir distorções nas várias modalidades

## 1.5 Objectivo

O objectivo desta tese é efectuar um levantamento das necessidades associadas ao co-registo visual da informação obtida através da electrocorticografia (ECoG) e da informação morfológica (RMI) do cérebro e implementar uma ferramenta para auxiliar neurocirurgiões e radiologistas na avaliação pré-cirúrgica, suportado por várias possibilidades de visualização.

Uma única ferramenta de visualização, que fundindo a informação eléctrica da ECoG e o posicionamento das grelhas de eléctrodos colocados na superfície do córtex, permite que a equipa constituída por neurofisiologistas, radiologistas e neurocirurgião consiga relacionar a morfologia do cérebro (ex. circunvoluções, fissuras), permitindo facilitar o planeamento da cirurgia.

Este trabalho é feito em colaboração com o programa de Cirurgia de Epilepsia do Hospital Universitário de Coimbra (HUC).

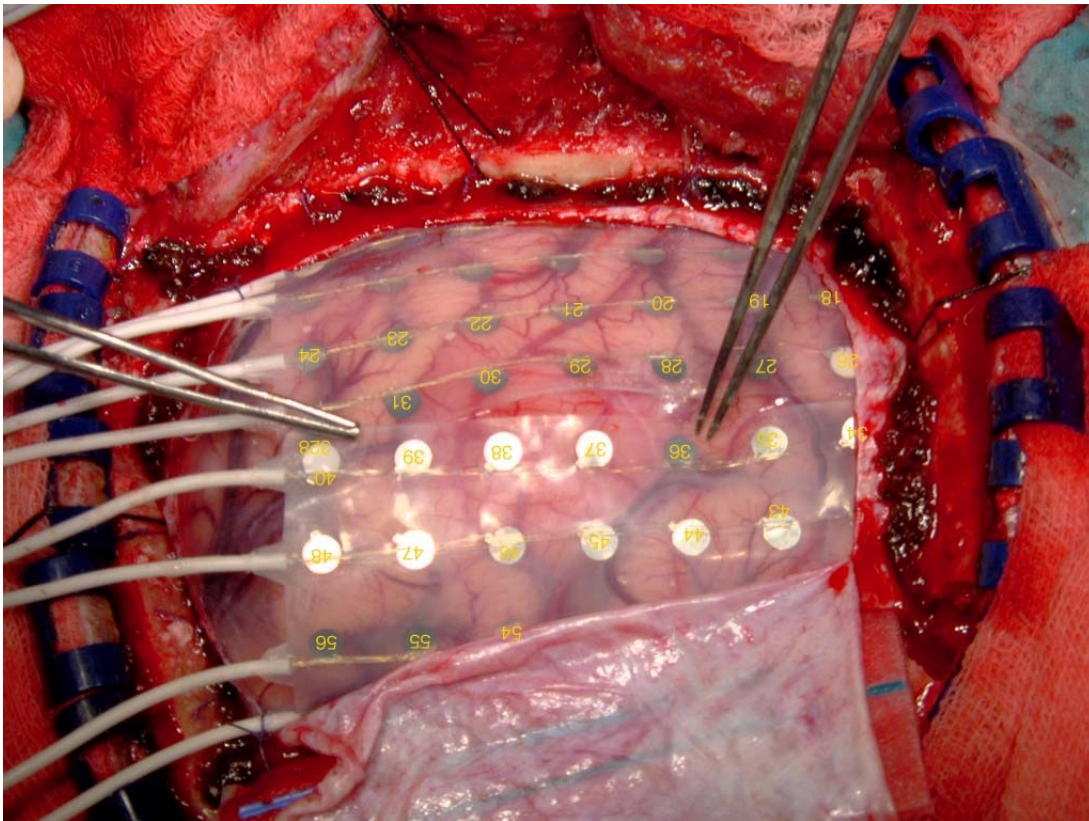


Figura 1.4: Colocação de uma grelha para ECoG

## Capítulo 2

# Enquadramento

### 2.1 Requisitos do co-registo visual de informação para avaliação invasiva pré-cirúrgica

Um dos objectivos desta tese é desenvolver uma plataforma que reúna as funcionalidades necessárias ao suporte ao planeamento cirúrgico suportado em electrocorticografia (ECoG). Esta plataforma deve permitir que as várias fontes de informação sejam co-registadas: a morfologia do cérebro (obtida por meio da RNM) e o sinal eléctrico intra-craniano como ilustrado no centro da figura 2.1.

#### 2.1.1 Dados

Do levantamento efectuado sobre os tipos de dados necessários à plataforma identificaram-se as seguintes fontes de informação:

- Sinal eléctrico resultante do ECoG - obtido através de eléctrodos dispostos numa grelha de eléctrodos colocada directamente no córtex do paciente.
- Esquema da grelha de eléctrodos - As localizações dos eléctrodos permite mapear a actividade eléctrica medida com a morfologia obtida através da RNM. Existem diversos modelos destas grelhas de eléctrodos, é necessário saber com qual se está a trabalhar e as suas especificações espaciais.
- Imagem de Ressonância Magnética - permite visualizar a morfologia do cérebro.
- Imagem de Tomografia Computadorizada - em contraste com a RNM, permite obter a localização exacta dos eléctrodos no córtex do paciente. A TC, ao contrário da RNM, não é afectada pela natureza metálica dos eléctrodos, logo permitindo a sua aquisição de forma segura - o que não acontece com a RNM.

#### 2.1.2 Processamento de Dados

Foi analisada a sequência de processamento necessário para a permitir a visualização e utilização dos dados no processo de planeamento pré-cirúrgico.





Figura 2.1: Diagrama Conceptual

- Inicialmente é necessário fazer o co-registo das imagens médicas das duas modalidades usadas, TC e RMI, de forma a garantir que os dados extraídos destas estão no mesmo referencial.
- É necessário de seguida obter as localizações precisas dos eléctrodos colocados na superfície do córtex. Este posicionamento é obtido a partir do TC usando um algoritmo de extração ou da marcação manual destes pontos sobre a imagem.
- Para poder obter a informação morfológica relevante, na visualização em 3D, implica recorrer a algoritmos de extração do crânio do cérebro/superfície cortical sobre a RMI, para ser possível a visualização dos eléctrodos previamente extraídos a partir do TC sobre a superfície do córtex.
- Para a visualização do sinal eléctrico é necessário um mapeamento entre os canais usados no ECoG eléctrodos extraídos da imagem de TC, agora mostrados sobre a superfície do córtex criada a partir da RMI segmentada.

Seguindo este fluxo de operações, é possível a visualização sincronizada de todos os dados.

## **2.1 Requisitos do co-registo visual de informação para avaliação invasiva pré-cirúrgica 15**

---

### **2.1.3 Visualização e Classificação**

Dada a relevância da visualização na utilização da plataforma, identificamos os seguintes requisitos a ela associados:

- visualização multimodal que forneça diferentes níveis de detalhe e permita variadas manipulações visuais.
- visualização em duas dimensões das fatias da imagem de RMI sincronizada combinando o posicionamento dos eléctrodos e a informação eléctrica proveniente da ECoG.
- visualização em três dimensões sincronizada que permita a um neurocirurgião a leitura do detalhe da morfologia do cérebro (circunvoluções, comissuras e outras) combinando também o posicionamento dos eléctrodos e a informação eléctrica do ECoG (ver nível superior do Diagrama Conceptual na figura 2.1).
- visualização da informação eléctrica pode ser representada por meio de diferentes níveis de intensidade por uma escala de cores, podendo assim ser mostrada sobre a representação da RMI, ou como é mais tradicionalmente apresentada, através de um gráfico (tempo por diferença de potencial). Para um maior rigor, ambos devem ser usados, um pelo seu valor como ferramenta de fácil interpretação, o outro como instrumento de rigor científico.
- integração dos eléctrodos com diversos marcadores textuais e visuais, como forma de classificação e marcação de zonas de interesse como é usual durante e após a interpretação dos resultados.

### **2.1.4 Solução e Requisitos**

Da análise anterior foram identificados os seguintes factores críticos para a aplicação:

- O co-registo das imagens médicas (CT com RMI).
- A extracção manual, semiautomática e automática das posições dos eléctrodos a partir do CT.
- Ferramenta de extracção do cortex.

Também fundamental é a visualização e classificação integrada após o tratamento dos dados:

- Visualização 2D e 3D da RMI, posições dos eléctrodos e os dados eléctricos da ECoG.
- Classificação dos eléctricos.
- Marcação dos eléctrodos, correspondência entre eléctrodos extraídos e os canais da ECoG

## 2.2 Casos de uso

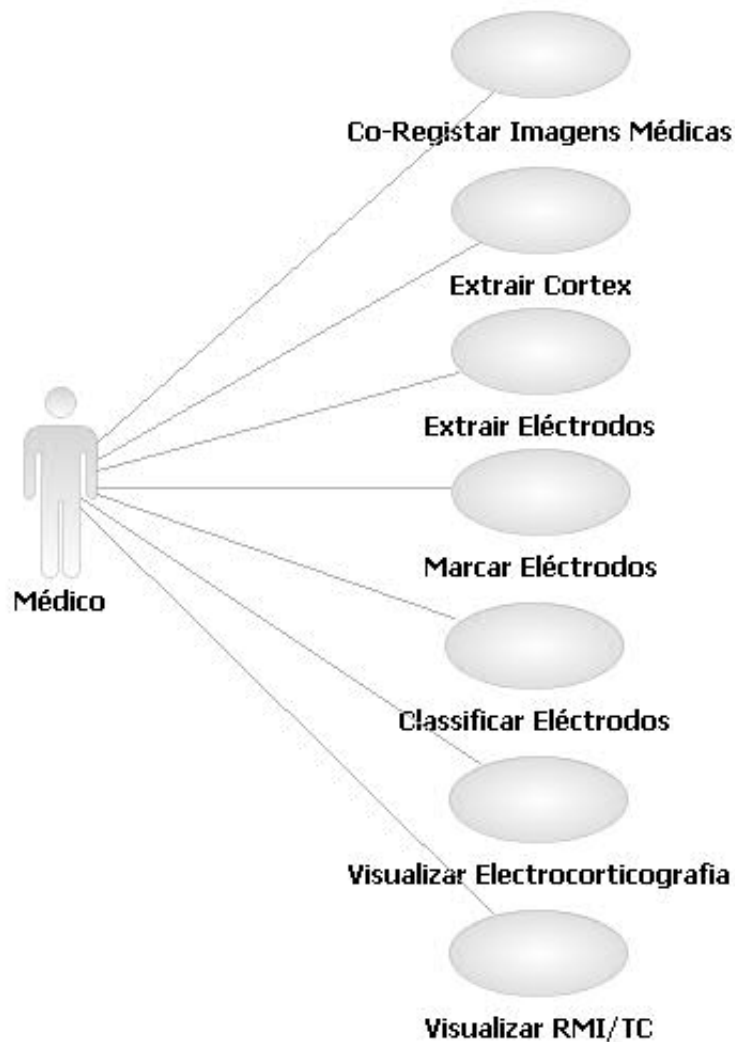


Figura 2.2: Visão funcional sobre o sistema

Foram isoladas também as funcionalidades mais relevantes necessárias à aplicação e traduzidas em vários casos de utilização.

No contexto da aplicação BrainGrid o Médico é visto como o actor principal que deve ser capaz de :

**Co-Registar Imagens Médicas** - permite ao médico efectuar o co-registo entre duas imagens médicas.

**Extrair Cortex** - permite ao médico extrair o córtex cerebral a partir da imagem de RM para melhor visualização da superfície do cérebro.

**Extrair Eléctrodos** - permite ao médico identificar facilmente numa imagem médica (CT neste

caso), pontos que correspondem a eléctrodos e marca-los para poderem ser usados posteriormente.

**Visualizar RMI/TC** - permite ao médico fazer uma visualização de imagens médicas de RMIs ou TCs quer como um volume 3D ou "fatias" podendo fazer vários ajustes.

**Visualizar Electrocorticografia** - permite a visualização dos dados eléctricos, tendo previamente extraído os eléctrodos, escolhido a grelha utilizada e os dados das medições desta, sobre a imagem da RMI ou CT, tendo assim os dados morfológicos e eléctricos sobrepostos ou apenas os dados eléctricos sobre a grelha de eléctrodos utilizada.

**Marcar Eléctrodos** - presentes em qualquer uma das formas de visualização, o médico pode atribuir-lhes diferentes cores para permitir uma maior diferenciação assim como

**Classificar Eléctrodos** - permite atribuir diferentes propriedades e etiquetas como que este está posicionado na zona da linguagem, memória, motora, ou sendo no início ou propagação da crise.

A figura 2.2 mostra uma visão funcional sobre o sistema

## 2.3 Aplicações de Visualização Médica

Foram estudadas várias aplicações de imagiologia médica e visualização em geral em busca de ideias e soluções para a plataforma.

Existem várias aplicações que permitem a visualização de informação médica, nomeadamente em alguns formatos mais populares na indústria - DICOM [dic, 2008] - e na área das imagiologia em neurociências - Analyze [ana, 1990]

Apresentam-se de seguida algumas dessas aplicações:

- SPM [spm, 2008]
- MRlcro [mri, 2008]
- VolView [vol, 2008]
- 3D Slicer [sli, 2008]
- EtDips [etd, 2008]
- ezDICOM [ezd, 2008]
- Julius [jul, 2007]
- AMIDE [ami, 2008]
- BrainSuite 2 [bra, 2008]

### SPM

A aplicação SPM [spm, 2008], baseada em MatLab, embora tendo como objectivo principal o processamento estatístico de imagem, fornece funcionalidades básicas e genéricas de pré-processamento que permitem não só a visualização como o co-registo de diferentes modalidades (e.g. fMRI, PET, SPECT, EEG e MEG).

### MRlcro

O MRlcro [mri, 2008] (ver Figura 2.3) é uma aplicação para visualização de imagem médica, permitindo a marcação de zonas de interesse (ROI) e co-registo visual de diferentes modalidades. Está especialmente optimizada para lidar com os formatos Analyze e NIFTI, fornecendo vários processos de conversão entre formatos de imagem médica.

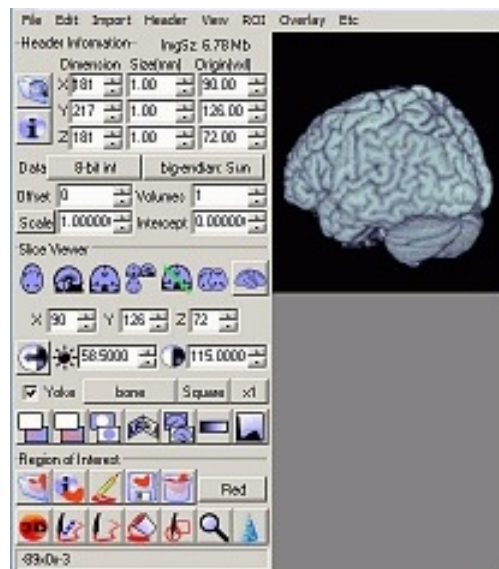


Figura 2.3: MRlcro

### VolView

O VolView [vol, 2008] (ver Figura 2.4) permite uma rápida exploração de imagens médicas 3D complexas ou imagens científicas. Permite a utilização de vários filtros, criação de contornos, medições, histogramas e anotações. Abre uma série de tipos de imagens (DICOM, Analyze, Visualization Toolkit, Raw, BMP, TIFF, PNG, JPEG, PNM e outros). É construído recorrendo às ferramentas de desenvolvimento da KitWare [kit, 2008]: VTK (Visualization Toolkit) e KWWIDGETS (GUI). Corre apenas em Windows.

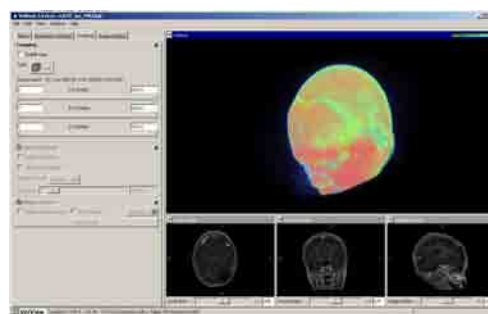


Figura 2.4: VolView

### 3D Slicer

3D Slicer [sli, 2008] (ver Figura 2.5) é um *software* de visualização, registo, segmentação e quantificação de dados médicos. Integra diversas potencialidades de processamento de imagens médicas num único ambiente. Fornece potencialidades ao nível da segmentação (extracção de superfícies), geração de modelos 3D, visualização 3D e análise quantitativa (medição de distancias, ângulos, áreas de superfície e volumes) de vários modalidades.

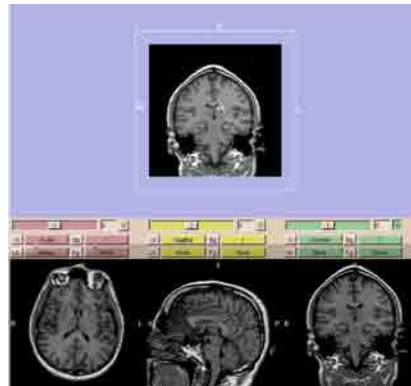


Figura 2.5: 3D Slicer

### EtDips

O *exploratory two/three dimensional image processing System* (EtDips) é um programa [etd, 2008] de visualização (ver Figura 2.6) e análise de volumes. Suporta vários tipos de visualização, e permite fazer render em tempo real. Aceita diferentes tipos de *input*, como DICOM, Analyze, e outros como Tiff ou Raw. E permite gravar as imagens em Analyze, Raw ou um formato proprietário. Tem também uma consola para operações matemáticas, filtros, aplicação de máscaras, mudanças de tamanho, etc.

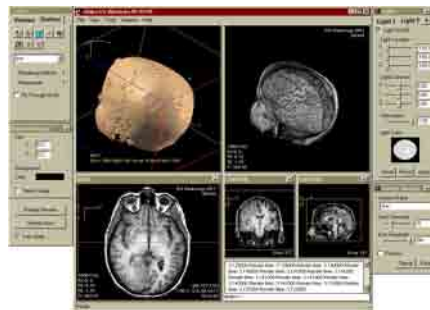


Figura 2.6: EtDips

### ezDICOM

Este *software* [ezd, 2008] (ver Figura 2.7) foi desenhado para a visualização da maioria das imagens médicas RMI, TC, raios X e ultra-sons. É de fácil utilização, estável e pode ler uma grande variedade de imagens médicas incluindo o formato proprietário DICOM. A maioria dos leitores livres do formato DICOM só lê uma pequena parte das imagens DICOM disponíveis. O ezDICOM também lê formatos como Analyze, GE (LX, Genesis), Interfile, Siemens (Magnetom, Somatom) e imagens NEMA. As suas maiores vantagens são ser livre, open source e ter boa documentação de suporte.

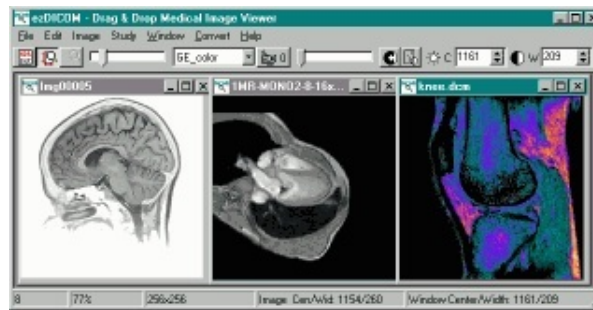


Figura 2.7: ezDICOM

### Julius

O *software* Julius [jul, 2007] (ver Figura 2.8) é uma framework para visualização de imagens médicas como o TC, RMI, SPECT e PET, e também uma ferramenta para desenvolvimento rápido de aplicações. O Julius Model Viewer é um exemplo de uma aplicação baseada nesta Framework. Permite a importação de imagens médicas e sua visualização e interação. A Framework permite a inclusão de diversos plugins para importação e exportação de vários formatos incluindo DICOM, visualização 3D, de superfícies e slices 2D. Usa a livreria gráfica Qt e pode ser usado livremente para projectos não comerciais.

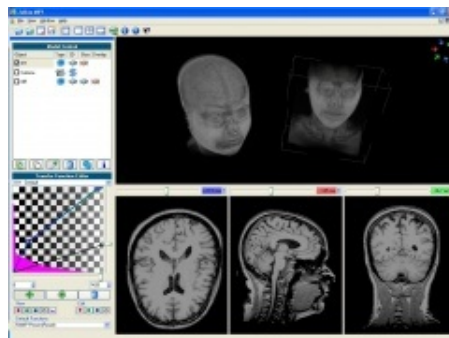


Figura 2.8: Julius

### AMIDE

AMIDE (Amide's a Medical Imaging Data Examiner) [ami, 2008] (ver Figura 2.9) é uma ferramenta para visualização, análise e co-registo de imagens médicas completamente livre. Foi desenvolvido usando GTK+ e corre em qualquer sistema operativo. Permite, entre outras coisas, importar vários formatos incluindo DICOM, definição de regiões de interesse, níveis de *threshold*, vários tipos de filtros, geração de filmes de navegação nas slices.



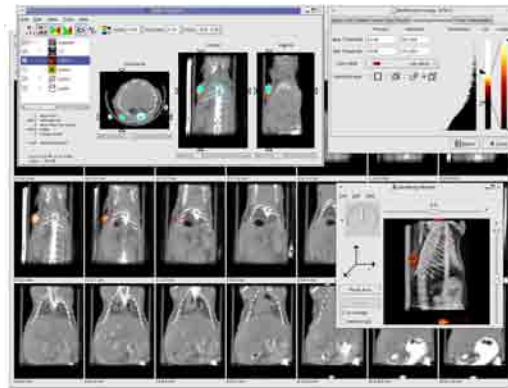


Figura 2.9: AMIDE

### BrainSuite 2

O BrainSuite2 é uma ferramenta de análise de imagens de Ressonância Magnética do cérebro desenhada para identificar tecidos e superfícies nestas imagens. O BrainSuite2 foi especialmente desenhado para guiar os utilizadores no processo da extracção da superfície cortical. O *software* foi escrito para requerer o mínimo de interacção com o utilizador para completar o processo de extracção da topologia esférica cortical da superfície de um volume de RM em poucos minutos. Os componentes individuais do BrainSuite podem também ser usados para segmentação de tecido mole, crânio e escalpe e para análise e visualização da superfície. Corre exclusivamente em Windows.

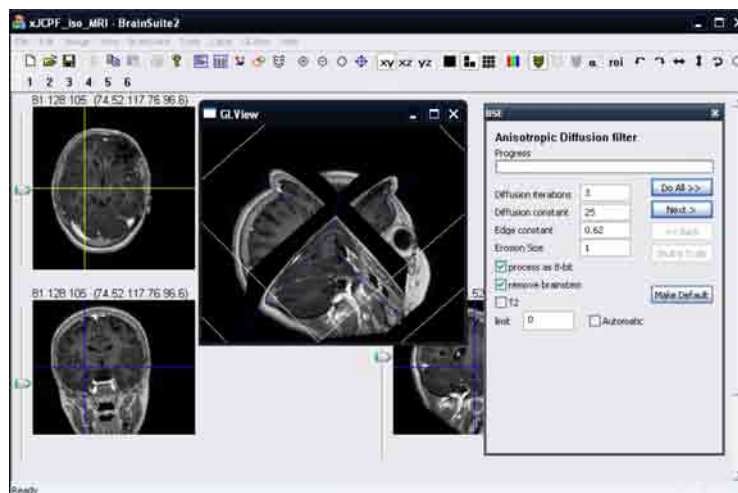


Figura 2.10: BrainSuite 2

## 2.4 Proposta

Da análise das soluções de interface das várias aplicações de visualização encontradas (secção 2.3, página 18) à luz dos requisitos identificados para o BrainGrid decidiu-se utilizar um interface que permitisse uma visualização multimodal interactiva, que possibilitasse uma vista 3D da imagem em questão bem como vistas 2D: axial, coronal e sagital (X,Y e Z), como utilizada por quase todas as aplicações revistas (MRlcro, VolView, 3D Slicer, EtDips, ezDICOM, Julius, BrainSuite 2).

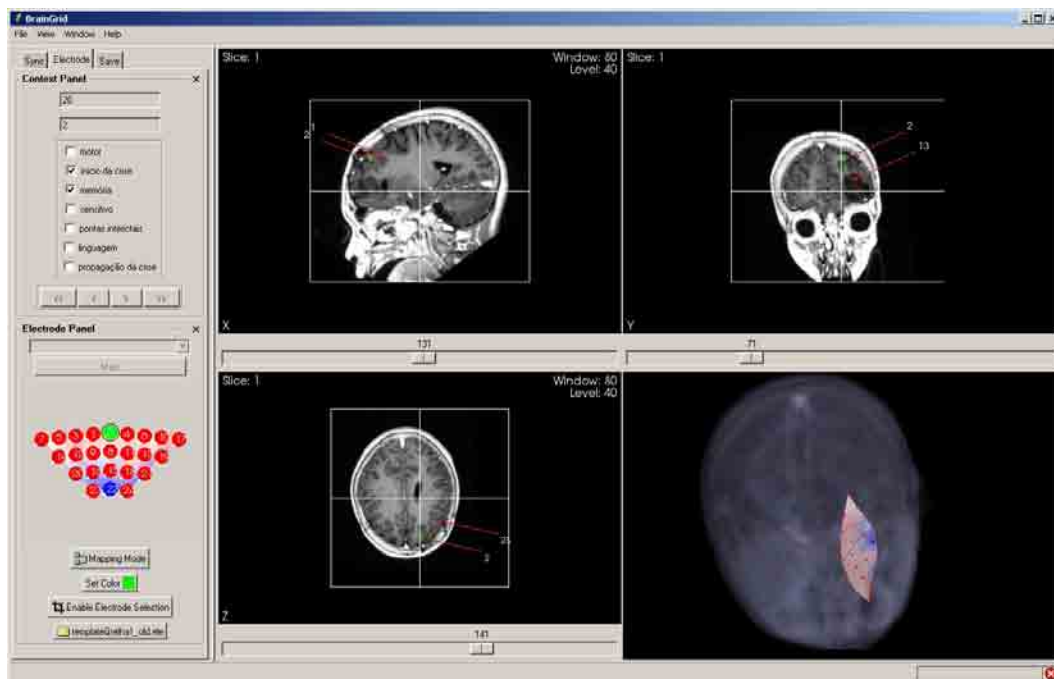


Figura 2.11: Proposta de Interface

À semelhança de algumas das aplicações revistas, deverá permitir fazer zoom nos vários modos de visualização, tanto através de botões de zoom predefinido (como no MRlcro, Figura 2.3), como utilizando o rato. Permitir translações e rotações na vista 3D.

Mas para o problema em questão, a electrocorticografia, esta aplicação (Figura 2.11) apresentará funcionalidades que a diferenciam de todas as outras aplicações revistas. A possibilidade de visualizar a grelha de eléctrodos utilizada na ECoG, sobre a imagem 3D (seja RMI ou CT) e sob a forma de um esquema representativo da mesma. Conter a possibilidade de parametrizar e lançar a extracção do cortex simplificada à semelhança do que faz o BrainSuite 2 (Figura 2.10). Deverá permitir uma marcação dos eléctrodos extraídos com os eléctrodos no esquema de grelha, de forma a estabelecer uma correspondência com os canais utilizados por cada um deles na electrocorticografia. Finalmente deve ainda, permitir a visualização da informação eléctrica ao longo do tempo, sobre a vista 3D e o esquema de grelha de eléctrodos.

## Capítulo 3

# BrainGrid - Co-registo Visual de Informação para Avaliação Pré-cirúrgica Invasiva

Neste capítulo vamos descrever a aplicação BrainGrid que pretende ser uma ferramenta para auxiliar neurocirurgiões e radiologistas na identificação das zonas do cérebro responsáveis pelas crises epilépticas, para melhor planeamento da cirurgia.

### 3.1 Organização da Aplicação

O diagrama da figura 3.1 explica de forma mais detalhada, o processo requerido para a utilização da nossa aplicação BrainGrid. Separa-se em 3 níveis:

- **Dados Iniciais** - partimos da imagem de Ressonância Magnética, da imagem de Tomografia Computadorizada, o esquema da grelha de eléctrodos usada para as medições e o sinal medido na ECoG.
- **Processamento dos dados** - a partir dos dados iniciais, temos o algoritmo de extracção do cérebro, o BET, do qual resulta uma imagem do cortex cerebral, que permite uma melhor visualização da morfologia em 3D. Este algoritmo pode ser parametrizado e corrido, a partir do BrainGrid. O co-registo, usando o ITK, para preparar a imagem do CT, que nos vai permitir ter a extracção das posições dos eléctrodos exactamente nas posições correctas relativamente à imagem do cortex a visualizar.
- **Visualização e Classificação** - é dentro na aplicação, que é feita a visualização 3D e 2D do cérebro, do sinal e dos eléctrodos. Assim como a classificação destes, como sendo de áreas responsáveis pelas funções nobres (memória, linguagem, etc) ou como zona de origem de uma crise. Anterior tendo o posicionamento dos eléctrodos, torna-se necessário fazer a marcação destes, associando cada eléctrodo no template da grelha com um extraído. Esta operação pode já ser realizada dentro no BrainGrid, podendo também ser feita a partir de um editor de texto.

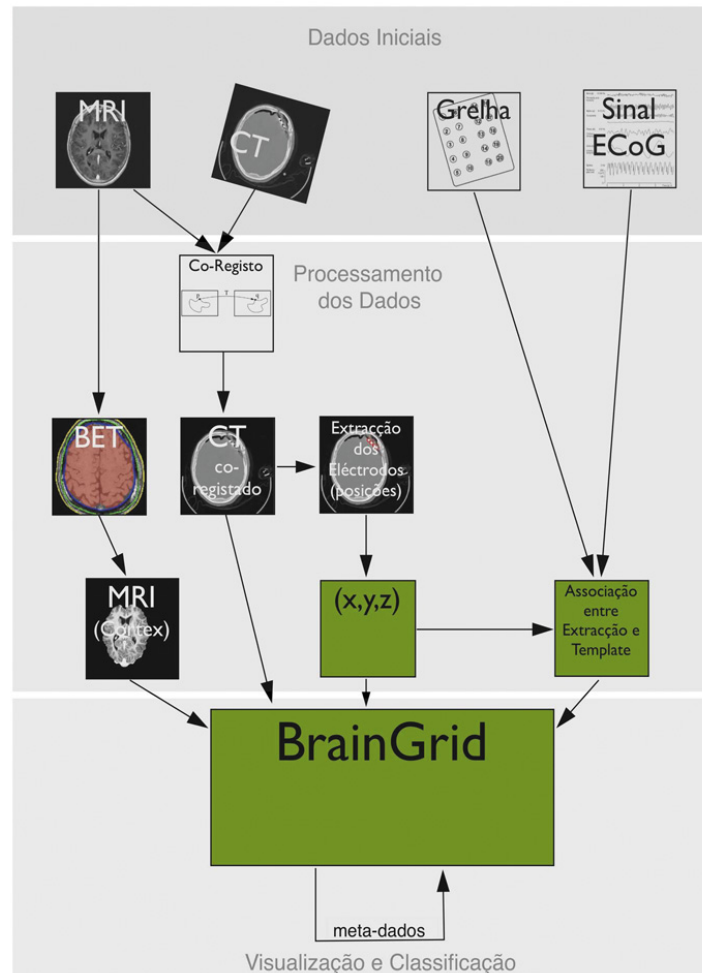


Figura 3.1: Diagrama da organização da aplicação

## 3.2 Escolhas Técnicas

Tendo sido delineados os requisitos funcionais e feita uma proposta de de interface, segue-se a escolha dos diversos suportes de *software* para a visualização, interface, co-registo e o formatos de imagem médica:

### 3.2.1 Visualização

A escolha da ferramenta a utilizar para efectuar a visualização recaiu sobre o VTK (Visualization Toolkit) [W. J. Schroeder, 1998, W. J. Schroeder, 2001].

O VTK é uma biblioteca livre para desenvolvimento rápido de sistemas de visualização, gráficos 3D e processamento de imagem. Consiste numa biblioteca de classes C++ e várias camadas de interfaces para Tcl/TK, Java e Python. Foi decidido fazer o desenvolvimento usando a linguagem em que o próprio VTK foi construído, C++. Esta biblioteca suporta uma série de algoritmos de

visualização que incluem métodos escalares, vectoriais, de tensores, texturas e volumétricos.

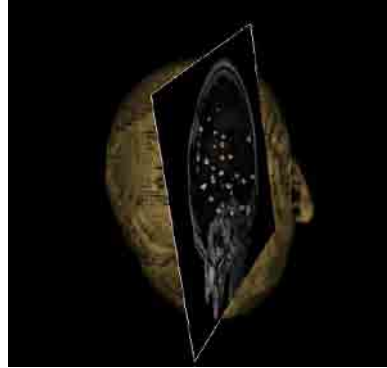


Figura 3.2: Exemplo VTK - Isocontouring com cortes

Suporta técnicas avançadas de modelação como redução de polígonos, triangulação de De-launay, contouring (ver figura 3.2), cutting, modelação implícita ou alisamento de malhas (*mesh smoothing*). Suporta ainda vários algoritmos de imagem que foram directamente integrados para permitir ao utilizador misturar imagem 2D com dados e algoritmos gráficos 3D. O desenho e implementação do VTK foram fortemente influenciados pelos princípios da orientação a objectos.

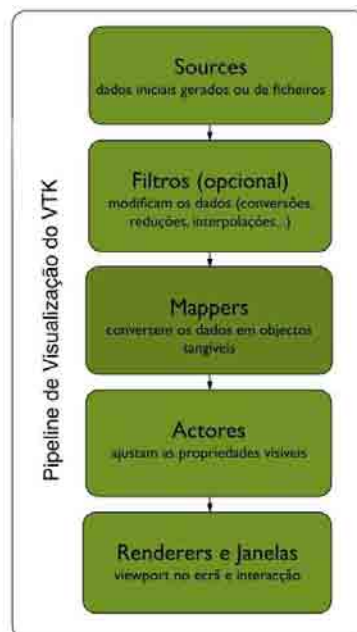


Figura 3.3: Pipeline de visualização do VTK

A estrutura central do VTK é uma pipeline de dados, desde a fonte de informação até à imagem renderizada no ecrã (ver figura 3.3). Tipicamente uma aplicação terá várias pipelines, correspondentes a vários itens que aparecem no ecrã. Estes podem ser diferentes representações do mesmo conjunto de dados inicial, diferentes selecções dos dados de entrada, objectos totalmente

independentes ou imagens.

É um sistema multi-plataforma correndo em Windows, sistemas Unix e Mac OSX.

Seria possível usar OpenGL [ope, 2008] para este projecto, mas rapidamente se percebe que seria necessário um tempo e esforço de desenvolvimento maior, dado que o VTK já integrava grande parte das ferramentas necessárias para atingir os objectivos propostos.

### 3.2.2 Processamento de imagem e Co-registo

O co-registo de imagens médicas é um processo complexo e não é o foco desta aplicação. Existem ferramentas que permitem construir mecanismos de co-registo mais ou menos específicos e eficazes.

O Insight Segmentation and Registration Toolkit (ITK) [itk, 2008] é uma *framework* de desenvolvimento de aplicações suportada por várias plataformas, muito usada para o desenvolvimento de programas de segmentação e co-registo de imagens.

O ITK foi desenvolvido com fundos da US National Library of Medicine of the National Institutes of Health a partir de 1999, como um recurso de algoritmos de código aberto para análise de imagens do Visible Human Project.

O *toolkit* contém alguns dos mais avançados algoritmos de segmentação e co-registo em duas ou mais dimensões. Está implementado em C++, sendo também possível o seu desenvolvimento em Tcl, Python e Java.

Algumas das funcionalidades deste toolkit são: a representação de dados e algoritmos para segmentação e co-registo cujo foco é em aplicações médicas apesar de ser capaz de processar outros tipos de dados. O suporte da representação de dados em forma geral para imagens de qualquer dimensão e redes não estruturadas (*unstructured meshes*). Não é dedicado à visualização ou interface gráfico, deixando isso a cargo de outros toolkits (como VTK, VisPack, 3DViewnix [vis, 2008, 3DV, 2008] estes últimos apenas para linux). Tem algumas ferramentas para interface gráfico mas muito básicas. E suporta o uso de threads para processamento paralelo.

Foi a nossa escolha como ferramenta de suporte ao co-registo sendo de notar ainda o grande número de filtros que podem permitir o estudo de outras aproximações interessantes tanto para o algoritmo de extracção de eléctrodos, como filtros simples.

### 3.2.3 Interface Gráfica - GUI

A interface gráfica com o utilizador ocupa uma parte muito importante do desenvolvimento de *software*. Por isso foram consideradas várias possibilidades:

- Qt [qt, 2008]
- wxWidgets [wxw, 2008]
- FOX [fox, 2008]
- KWWidgets [kww, 2008]

#### Qt

O Qt [qt, 2008] é uma framework multi-plataforma para desenvolvimento de aplicações, é muito usada para o desenvolvimento de GUI's (Graphical User Interface) apesar de poder ser usada para aplicações sem GUI (para consola ou para servidores). A sua utilização é visível em aplicações como KDE, o browser Opera, o Google Earth ou o Skype. A maior restrição encontrada, prendia-se com a sua licença, apesar de ser livre para uso pessoal, é restrita a nível comercial.

#### wxWidgets

O wxWidgets [wxw, 2008] é uma biblioteca com os elementos básicos para a construção de interfaces gráficas com o utilizador, ligação a bases de dados ODBC e ligação por sockets. É multi-plataforma, livre tanto para uso pessoal como para uso comercial e o código é aberto. O seu desenvolvimento não é somente feito em C++, mas também tem interfaces para python, perl, java, lua, eiffel, C#, .NET, basic, ruby e até javascript.

#### FOX

O FOX [fox, 2008] é um toolkit baseado em C++ para desenvolvimento fácil e eficaz de interfaces gráficas, também multi-plataforma e livre. Oferece uma extensa colecção de controlos assim como interfaces para *widgets* OpenGL para manipulação gráfica 3D. Foi dada uma importancia considerável a tornar o FOX num dos toolkits mais rápidos disponiveis assim como de minimizar o uso de memória.

#### KWWidgets

O KWWidgets [kww, 2008] é uma biblioteca de código aberto, livre e multi-plataforma de classes de GUI baseado em Tcl/Tk com um interface para C++. Esta biblioteca foi desenvolvida originalmente pela KitWare para a sua aplicação ParaView, e posteriormente foi extendida em funcionalidade e arquitectura graças ao suporte da NAMIC [nam, 2008] (National Alliance for Medical Image Computing). Foi construido à semelhança do VTK e para ser usado com o VTK pela empresa que desenvolveu o VTK.

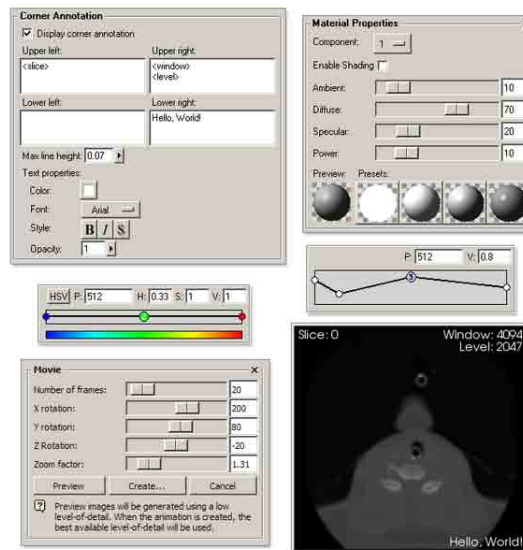


Figura 3.4: Algumas *widgets* do KWWidgets

Escolhemos o KWWidgets, pois tem um boa integração com VTK e foi desenvolvido com intenção de ser um toolkit de apoio a ferramentas de visualização, nomeadamente visualização médica (ver 3.4).



### 3.2.4 Formatos Médicos

Um dos maiores problemas com as imagens médicas é seu acesso. A variedade de formatos de armazenamento e por consequência a necessidade de aceder aos seus conteúdos em formatos proprietários, o que dificulta sua integração com imagens de outros aparelhos, ou sua reconstrução. Este problema levou ao aparecimento de alguns formatos médicos para serem usados como standard.

#### Analyze

O formato Analyze [ana, 1990] começou a ser desenvolvido no início dos anos 70 pelo Biomedical Imaging Resource/Mayo Clinic para ser aplicado numa vasta gama de visualizações bio-médicas e científicas. Este formato tem sido usado para visualizar e manipular imagens médicas a partir de modalidades como TC, RMI, SPECT e PET usando o *software* SPM para análise de dados. Este formato consiste em dois ficheiros, um para o cabeçalho e outro para os dados da imagem. O ficheiro do cabeçalho tem a extensão \*.hdr e contém informação sobre o ficheiro de dados como marcas para a representação do volume através de imagens ou numero de pixéis em cada eixo coordenado. O ficheiro que contém os dados tem a extensão \*.img e é constituído por dados binários que representam o stream de intensidade de cada voxel. Este formato tem a desvantagem de não suportar qualquer tipo de compressão dos dados sem perdas, pelo que os mesmos apresentam dimensões muito grandes sobrecarregando o processamento dos mesmos. O formato Analyze é muito usado no campo da neuroimagem funcional e mais recentemente surgiu também o NIFTI (Neuroimaging Informatics Technology Initiative) [nif, 2008], uma evolução open source do formato Analyze.

#### DICOM

O standard DICOM [dic, 2008] (Digital Imaging and Communications in Medicine) foi criado originalmente pela NEMA (National Electrical Manufacturers Association) para ajudar na distribuição e visualização de imagens médicas, como Tomografias Computadorizadas (CT), Resonâncias Magnéticas (RMI) e ultra-sons. Este formato é uma extensão do antigo formato standard na NEMA. A maioria das pessoas refere-se às imagens compatíveis com a parte 10 do standard DICOM como ficheiros no formato DICOM. Um único ficheiro DICOM contém um cabeçalho, (que guarda informação sobre o nome do paciente, o tipo da modalidade, as dimensões da imagem e outros) e todos os dados da imagem (que guarda a informação da imagem a três dimensões). Este é diferente do formato popular, Analyze que guarda os dados num único ficheiro (\*.img) e o cabeçalho noutro ficheiro (\*.hdr). Outra diferença entre estes formatos, é que o formato DICOM permite a compressão (encapsulada) para reduzir o tamanho da imagem. Os ficheiros podem ser comprimidos usando diferentes variantes do formato JPEG, assim como o RLE (Run-Lenght Encoding format). É o standard mais popular nos fabricantes de máquinas de modalidades imagiológicas como por exemplo RM, TC, SPECT entre outras, daí a sua relevância. Caso seja necessário usar o SPM para normalizar as imagens pode ser necessário converter este formato para o formato Analyze.

Decidiu-se escolher como prioritário o uso do formato Analyze, pois este é mais fácil de ler e trabalhar. Não se exclui no entanto o uso do formato DICOM quando necessário.

### **3.2.5 Plataforma**

Inicialmente foram consideradas tanto a plataforma Microsoft Windows como a plataforma Linux. Foram feitos alguns testes em ambas (Microsoft Windows XP, 2000 e Ubuntu 6.06), mas dada a dificuldade de colocar em funcionamento por completo, alguns dos toolkits considerados na plataforma Linux, a opção recaiu sobre a plataforma Windows. Com a vantagem acrescida de ser a plataforma usada para anteriores projectos de visualização na Universidade. E também de algumas das mais importantes aplicações de referência estudadas.

Verifica-se no entanto que existem mais iniciativas de visualização médica livres e de código aberto, com resultados igualmente interessantes na plataforma Linux.

### 3.3 A Aplicação

#### 3.3.1 Arquitectura

A arquitectura da aplicação pode ser representada por 4 níveis distintos:

- Tratamento dos Dados
- Visualização
- Interface (GUI)
- Dados

O pacote de Tratamentos dos dados contém as classes responsáveis pelo co-registo das imagens TC com a RM e para extração de eléctrodos de imagens TC. O pacote de Visualização contém as classes responsáveis pela leitura dos dados e sua consequente visualização. São as classes deste pacote, que geram as imagens 3D e 2D que vão ser apresentadas pelas classes do pacote GUI. As classes do pacote GUI são responsáveis pela criação das janelas da aplicação e *widgets* de controlo. O pacote Dados contém as estruturas onde são guardados os dados dos eléctrodos necessários à visualização e os dados do template da grelha de eléctrodos. A Figura 3.5 representa as classes que contém ligações a outros pacotes e essas ligações.

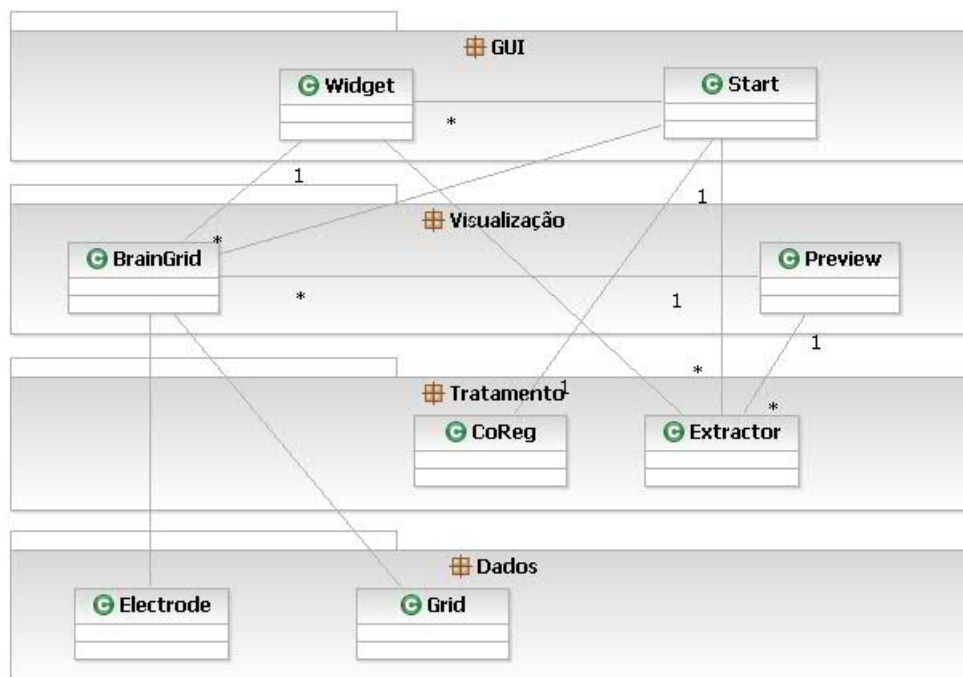


Figura 3.5: Arquitectura

### Vista Geral

Através da vista geral apresentada na Figura 3.6, pode ter-se visão mais abrangente do sistema, incluindo os pacotes auxiliares na visualização. Uma descrição mais detalhada de cada um destes pacotes será feita nas secções seguintes. As classes estão representadas sem métodos nem atributos para simplificar a sua visualização.

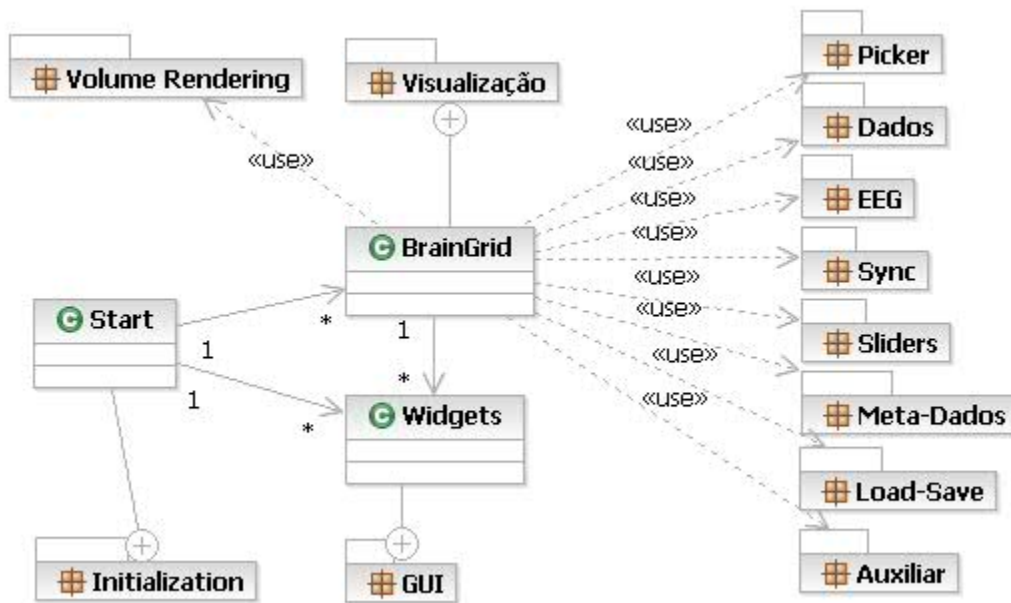


Figura 3.6: Vista Geral do diagrama de classes

Encontram-se no anexo A (página 70), a descrição dos diversos pacotes apresentados na vista geral do diagrama de classes, nomeadamente os pacotes de: Visualização, Dados, Load/Save, ECoG, Inicialização, Tratamento de Dados, GUI, Auxiliar, Navegação de Meta-Dados, Sliders, Sincronização e Pickers.

Nas secções seguintes serão descritos os vários pontos de domínio técnico da aplicação:

- Leitura
- Visualização 3D
- Visualização 2D
- Co-Registo
- Segmentação do Cortex
- Extracção de eléctrodos
- Visualização de eléctrodos
- Templates de grelha
- Sincronização
- Visualização dos dados eléctricos
- Eléctrodos: propriedades
- Interface com o utilizador

### **3.3.2 Leitura**

Para os vários casos médicos estudados, foram usadas imagens em formato Analyze. Este formato consiste em dois ficheiros, um para o cabeçalho e outro para os dados da imagem. O ficheiro do cabeçalho tem a extensão \*.hdr e contém informação sobre o ficheiro de dados. O ficheiro que contém os dados tem a extensão \*.img e é constituído por dados binários que representam o stream de intensidade de cada voxel.

Incluindo o cabeçalho dbh.h do formato Analyze [ana, 1990] basta fazer a leitura do ficheiro para essa estrutura, e assim teremos guardada toda a informação relativa às dimensões e espaçamento necessárias.

Para ler o ficheiro que contém a imagem em si, foi usada a classe `vtkImageReader2`. Com as informações obtidas a partir do cabeçalho, facilmente são preenchidos os parâmetros necessários (*Extent* e *Spacing*).

### **3.3.3 Visualização 3D**

Para visualizar os dados de imagens médicas, seja TC, RMI, ou outra num formato legível foram testados vários métodos de visualização, mas especialmente o método de Volume Rendering pois apesar de ser o método computacionalmente mais intensivo é o que nos dá um melhor detalhe da imagem. Este método consiste na visualização de um volume de dados composto por amostras de várias densidades ou voxels sem a definição explícita de uma superfície geométrica. O pipeline implementado está representado na figura 3.7.

Depois de imagem ser lida pelo classe `vtkImageReader`, é usada a classe `vtkImageResample` para re-amostrar para ficar maior ou mais pequena, conforme o pretendido, neste caso para

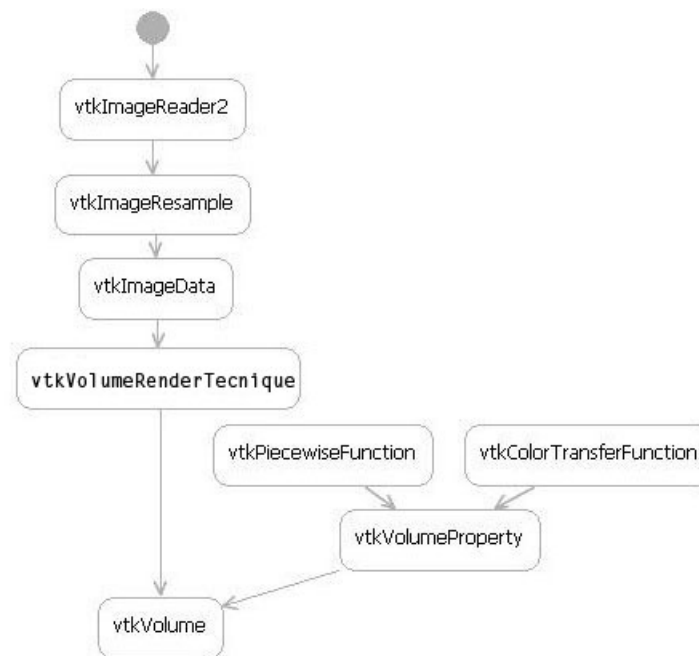


Figura 3.7: Pipeline de Visualização 3D

melhorar a performance. O output desta filtragem é um `vtkImageData`, *input* da classe `vtkVolumeRayCastMapper`, que estabelece o interface entre os dados e as primitivas gráficas ou técnicas de rendering.

É criado um `vtkVolume` cujo *mapper* vai ser a classe `vtkVolumeRayCastMapper` e cujas propriedades são definidas no `vtkVolumeProperty` através das classes `vtkPiecewiseFunction` e `vtkColorTransferFunction`, a primeira define uma função de mapeamento seccionalmente linear para ser usada na segunda (função de transferencia de cor). Um bom resultado da visualização prende-se muito com os parâmetros usados para as funções transferencia de cor e opacidade, assim como o tipo de interpolação (linear ou de proximidade) e também as propriedades de reflexão de luz do volume. Na figura 3.8 ficam alguns dos testes feitos com diferentes parâmetros.

Esse Volume (`vtkVolume`) é adicionado posteriormente a ao renderer (`vtkRenderer`) para ser visualizado. Para agilizar o processo de encontrar a melhor parametrização para cada imagem, foram criados uma serie de *widgets* no interface, a ver em pormenor na secção 3.3.19.

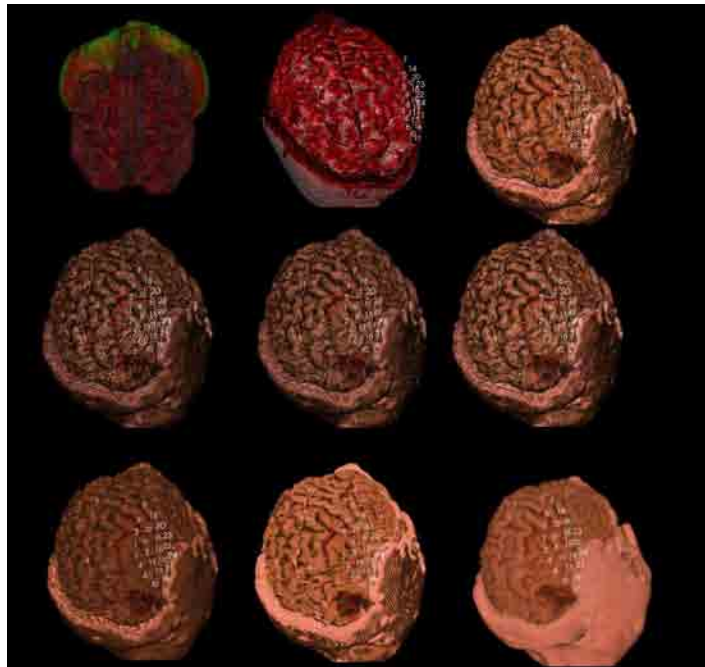


Figura 3.8: Testes feitos com a tecnica Volume Ray Cast, variando a função transferencia de cor, opacidade, e propriedades de luz do volume

### 3.3.4 Visualização 2D

Para poder fazer a visualização dos cortes axial, coronal e sagital foi utilizada a classe `vtkImagePlaneWidget`. É criado um destes *widgets* para cada uma das vistas acima indicadas, diferenciados pela orientação com que cortam a imagem (X, Y e Z). Cada um associado a um `vtkImageReader` e a cujo output vai ser por sua vez apresentado num `renderer` (`vtkRenderer`) (ver figura 3.18).

### 3.3.5 Co-Registo

Como prova de conceito, foi adaptado um algoritmo de co-registo, que faz parte do conjunto de ferramentas do BrainGrid, podendo o co-registo ser realizado externamente recorrendo a outras ferramentas como o SPM [spm, 2008] ou o FSL [fsl, 2008]. Os componentes da framework de co-registo e as suas interligações são mostrados na figura 3.10. O *input* básico do processo de co-registo são duas imagens, uma definida como *fixed image* (de referência) e a outra como *moving image*. O co-registo é tratado como um problema de optimização com o objectivo de encontrar o mapeamento espacial que alinha a *moving image* com a *fixed image*.

O componente da transformação representa o mapeamento espacial dos pontos do espaço da *fixed image* com os pontos do espaço da *moving image*.

A componente da métrica dá uma medida de quão precisamente a *fixed image* está co-registada com a *moving image*. Esta medida forma o critério quantitativo para ser optimizado pelo optimizador na procura do espaço definido pelos parâmetros da transformação. O co-registo

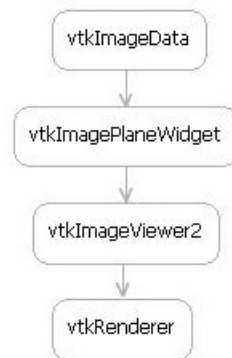
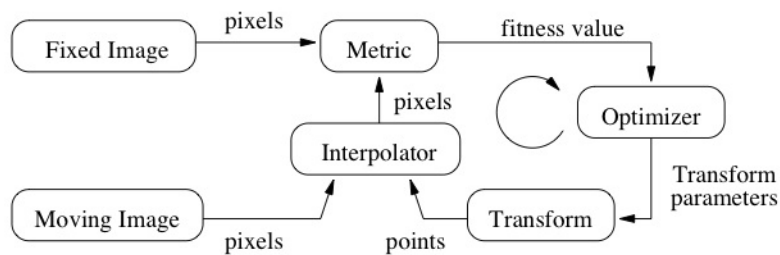


Figura 3.9: Pipeline de Visualização 2D

Figura 3.10: Componentes básicos da framework de co-registo: as duas imagens, a transformação, a métrica, o interpolador e o otimizador (*Fonte: The ITK Software Guide*)

é atingido pelo ajuste da posição e orientação relativas até a informação mutua ser maximizada. No nosso caso foi adaptado um dos templates de co-registo do ITK:

`Examples/Registration/ImageRegistration8.cxx`.



### 3.3.6 Segmentação do Cortex

Existem várias aplicações relacionadas com a imagiologia do cérebro, que necessitam de segmentação de cortes do volume craneano com precisão. Por exemplo, no co-registo de imagens funcionais como o PET com imagens de RM[Smith, 2002]: como as primeiras contêm pouco tecido não cerebral, e as imagens de RM de alta resolução provavelmente contêm a representação dos olhos, pele, gordura, músculo, o co-registo pode ser facilitado pela remoção prévia destes elementos. Outro exemplo da utilidade da segmentação do cérebro é de permitir a remoção de certos elementos que vão possibilitar uma melhor visualização da superfície do cérebro, como é o caso neste projecto.

Até à data, os métodos para a segmentação cérebro/não cérebro inserem-se em três categorias distintas: manual, *thresholding-with-morphology* e baseados em modelo de superfície.

Os métodos de segmentação manuais envolvem uma compreensão da informação complexa envolvida, são normalmente mais precisos que métodos completamente automatizados. Normalmente os métodos manuais, não são os mais viáveis. Em primeiro lugar porque o custo temporal da segmentação manual cérebro/não cérebro é de cerca de 15 minutos a 2 horas por volume. Em segundo lugar é necessário treino suficiente e alguma dedicação para a precisão e o custo temporal seja de um nível aceitável.

Foram estudados alguns testes comparativos já feitos pelo FMRIB (Oxford Center of Functional Magnetic Resonance Brain Imaging)[Smith, 2002], entre dois métodos de *thresholding-with-morphology*, o BSE (Brain Surface Extraction) e o método incluído no AFNI (conjunto de ferramentas para análise de imagens funcionais do cérebro) e um método desenvolvido pelo próprio FMRIB, o **BET (Brain Extraction Tool)**, baseado em modelos de superfície.

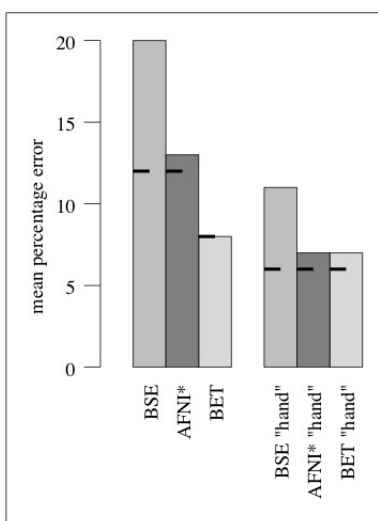


Figura 3.11: Comparação entre BET, AFNI e BSE.

Comparação entre BET, AFNI e BSE (Fonte: *FMRIB technical Report TR00SMS2b*). O gráfico mostra a média % do erro dos três métodos extracção do cérebro, comparados com as segmentações manuais (de referência) para 45 imagens de RM. À direita estão os resultados dos testes que foram optimizados à mão

Os resultados são mostrados na figura 3.11. É de notar, que não só o BET é o mais preciso e robusto dos vários métodos em ambos os testes, é também o algoritmo que tem melhores resultados quando o processo é completamente automatizado, quase tão bons como os otimizados manualmente.

O método do BET consiste em inicialmente, processar o histograma de intensidades para encontrar um limite superior e inferior de valores de intensidade para a imagem e um limiar cérebro/não cérebro inicial. O centro de gravidade da imagem é encontrado assim como um o valor inicial do tamanho da cabeça. De seguida é feita uma triangulação de uma esfera que é colocada em torno do centro de massa, que é lentamente deformada um vertex de cada vez, seguindo as direcções que mantêm a superfície bem balanceada e suave, tentando mover-se para o limite do cérebro. Se não é atingido um resultado satisfatório nesta fase do processo, ele é repetido com maiores restrições de *smoothness*. Finalmente é necessário que seja estimada a superfície do crânio. Todo este processo é ilustrado na Figura 3.12. Foi usado o binário incluído com o programa MRicro, executado a partir do BrainGrid.

Este método é muito robusto e preciso e foi testado em milhares de imagens de uma grande variedade de scanners. O BET demora entre 5 e 20 segundos numa workstation moderna e é livremente distribuído como parte do FSL (FMRIB Software Library) [fsl, 2008].

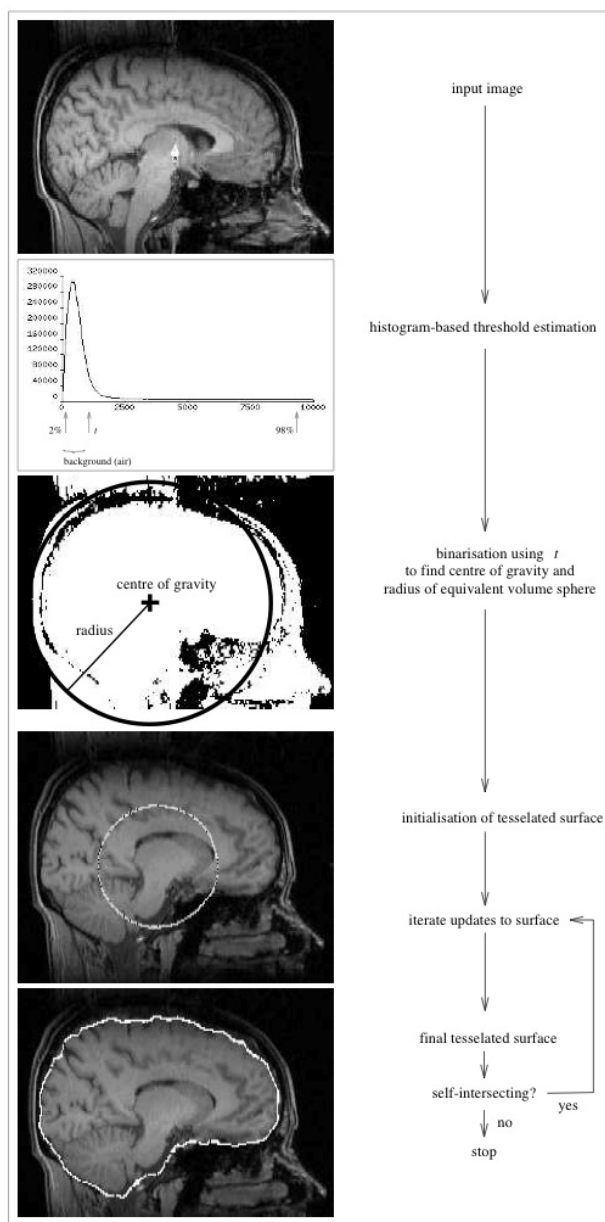


Figura 3.12: BET *flowchart* [Smith, 2002] (Fonte: FMRIB technical Report TR00SMS2b)

### 3.3.7 Extracção de eléctrodos

#### Automática

A extracção dos eléctrodos é feita a partir de uma Tomografia Computadorizada. Notou-se que os eléctrodos na imagem têm níveis de cor entre os 220 e 255 numa escala de cinzentos. Para poder saber a sua localização procurou-se marcar os centros das manchas com esses níveis de cor. Primeiro é afinado o intervalo dos níveis de cor, e usando a classe `vtkImageThreshold`, cujo *input* é um `vtkImageData`, pondo assim a preto (nível de cor = 0) todos os voxels fora do intervalo definido e a branco (nível de cor = 255) os voxels dentro do intervalo. Seguidamente é feita uma pré-Filtragem (Pre-Filtering na Figura 3.24) onde se procura na vizinhança de cada voxel por voxels a branco, caso este esteja isolado (nenhum voxel a branco) é posto a preto. De seguida, cada uma das manchas é marcada com uma semente (*seed*), para servir de referencia para o `vtkImageSeedConnectivity`, que tem como *input* a `vtkImageData` resultante do Pre-Filtering, para calcular os centros das manchas (ver pipeline na Figura 3.13) .

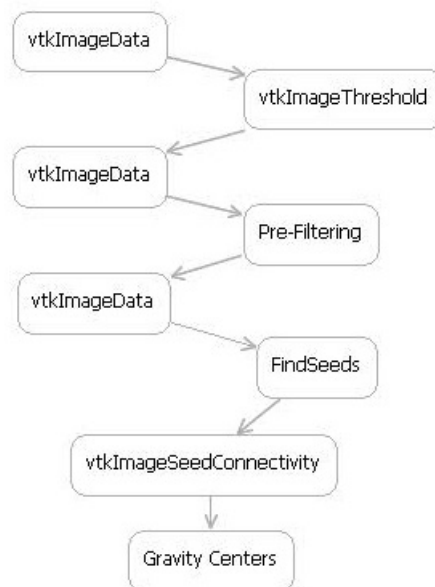


Figura 3.13: Algoritmo de extracção de eléctrodos

Cada um dos centros das manchas (*Gravity Centers*), é depois representado por uma esfera ( `vtkSphereSource` - `vtkMapper` - `vtkActor`) e colocado no renderer (`vtkRenderer`). A imagem da Tomografia Computadorizada original é também representada de forma a ser possível uma marcação mais eficaz dos eléctrodos. Cada esfera poderá ser marcada (através de um *callback* de *picking*) com uma cor diferente para assinalar ser efectivamente um eléctrodo, depois de terminado este processo, as posições dos eléctrodos podem ser escritas para um ficheiro. Este processo será clarificado noutra secção.

## Manual

Como em muitos casos, o algoritmo anteriormente descrito, não é o ideal. Foi construída uma ferramenta para a extracção manual de eléctrodos. Tem a vantagem de ser mais flexível, poder ser usado em todos os casos e com imagens com menor qualidade.

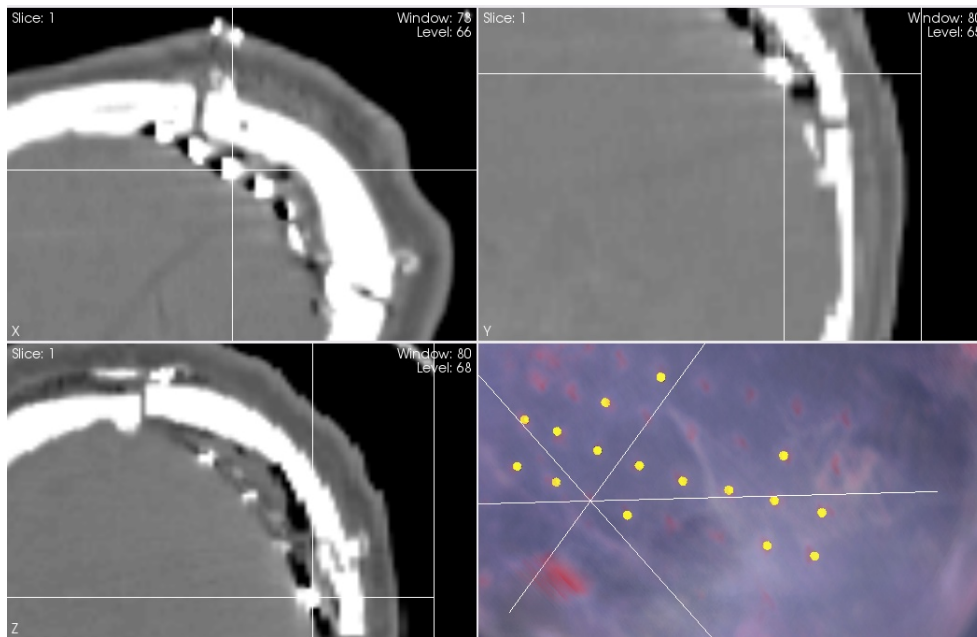


Figura 3.14: Vistas 2D e 3D do interface de extracção manual.

Na base do sistema manual está a possibilidade de uma visualização multimodal, onde se pode adaptar a visualização 3D para permitir ver as posições dos eléctrodos e adaptar também as vistas axial, coronal e sagital ao nível de zoom e contraste para a selecção dos pontos correspondentes aos electrodos. Esta selecção é feita através do click numa das 3 vistas a 2D, e afinadas através do uso dos sliders. Isto graças ao sistema de sincronização explicado na secção 3.3.10.

A partir daqui, cada posição pode ser adicionada a uma lista de pontos (ver Figura 3.15), onde podem ser removidos se necessário. Posteriormente estes pontos podem ser escritos para um ficheiro para poderem ser usados na nossa aplicação de visualização.

Id	X	Y	Z
4	114.81	148.71	191.1
5	114	139.42	194.67
6	115.5	157.96	185.28
7	115.5	167.35	180.08
8	114	174.44	173.46
9	114	184.16	153.14
10	111	181.44	167.23
11	111	191.33	146.46
12	108	187.87	160.56
13	102	191.83	167.58

Figura 3.15: Lista de pontos no interface de extracção manual.

### 3.3.8 Visualização de eléctrodos

As posições dos eléctrodos previamente extraídos são guardadas num ficheiro. A partir dessas posições, vai ser criado um vector de estruturas “Electrode”, que guarda informação relativa a aos eléctrodos (ID, posição, tipo) . Para cada um dos eléctrodos, vai ser usada uma esfera, para o representar tanto na vista 3D como nas vistas 2D, e vai ter associado um label que indica o seu ID, como se pode ver na Figura 3.18. Para o caso das vistas 2D, por slices, apenas os eléctrodos cuja posição é coincidente a slice em visualização vão ser mostrados. Cada vez que há uma movimentação da slice (através do uso do slider, planos de corte, ou devido a qualquer sincronização), são retirados todos os eléctrodos e seus respectivos labels, e é feita a verificação de quais os eléctrodos a ser mostrados, que vão ser colocados também obedecendo ao esquema da Figura 3.16.

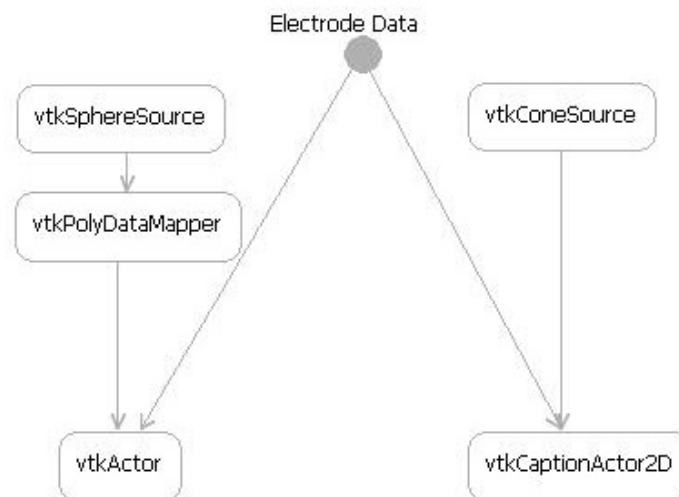


Figura 3.16: Pipeline de colocação de eléctrodos

### 3.3.9 Templates de grelha

Para permitir maior facilidade de leitura da informação e sincronização, existe uma vista do template da grelha. Existem vários tipos de grelhas e o template usado difere de caso para caso. De forma a permitir uma maior flexibilidade é usado um ficheiro que descreve a estrutura do template. Cada eléctrodo do template é descrito por uma linha que contém informação relativa ao seu id, posição, referencia ao eléctrodo real e classificação. Obedece à estrutura seguinte:

```
(id, referência ao eléctrodo real, x, y, R, G, B,  
label0, label1, label2, label3, label4, label5, label6);
```

É possível assim adaptar a nossa aplicação a qualquer tipo de grelha, como ilustrado na Figura 3.17.

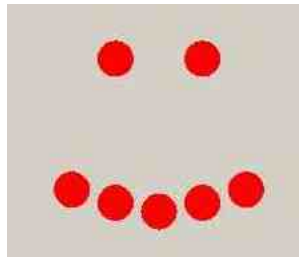


Figura 3.17: Exemplo de template de grelha (fictícia)

### 3.3.10 Sincronização

A sincronização entre janelas é fundamental para gestão da visualização da informação. É necessário a sincronização entre a vista 3D e as vistas 2D, e também com a vista de grelha e os meta-dados do eléctrodo. Relativamente à sincronização entre vistas 2D, quando ocorre um click sobre um ponto da imagem de qualquer uma das vistas, as outras serão alinhadas consoante a posição seleccionada, como na Figura 3.18.

Para uma maior precisão, é usada uma mira para referenciar a posição seleccionada. Para o fazer é usado um `vtkAbstractPicker`, obtido a partir do `vtkRenderWindowInteractor`, e a posição sobre a qual foi feito o click a partir deste e actualizar assim os outros viewers.

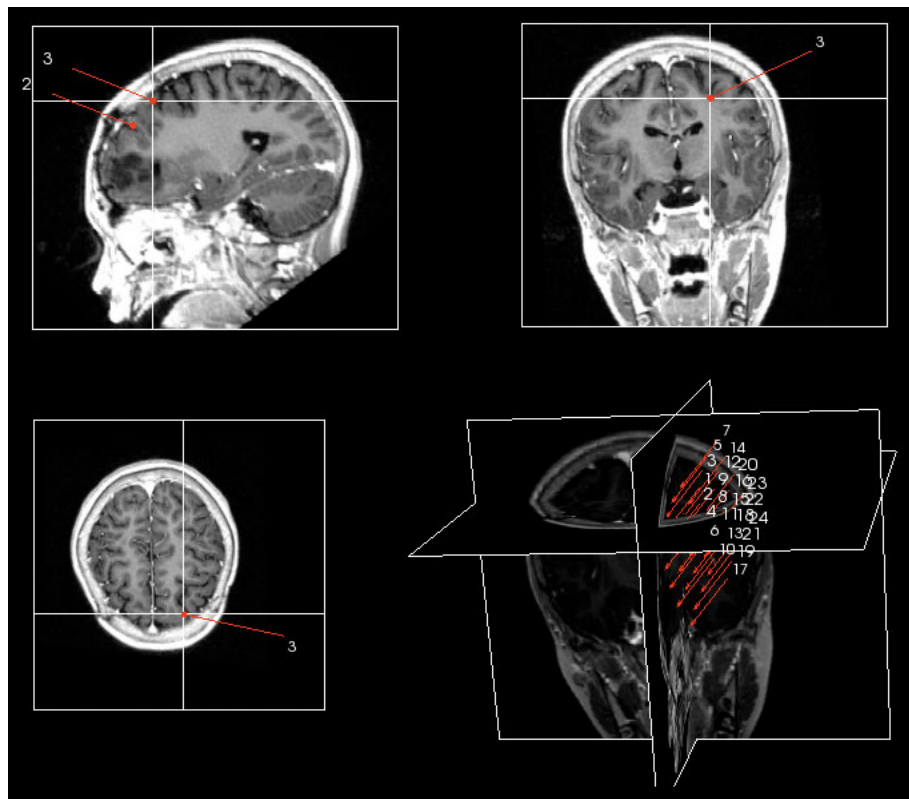


Figura 3.18: Vista 3D e vistas 2D sincronizadas.

Quando ocorre um *click* sobre um eléctrodo na vista 3D, as vistas 2D são alinhadas pela posição desse eléctrodo. O mesmo ocorre quando é feito um click sobre um eléctrodo na vista de grelha de eléctrodos. Se para a vista 3D, é apenas necessário alinhar as vistas 2D com base na posição do eléctrodo seleccionado, para a vista de grelha, que é apenas ilustrativa da posição relativa dos eléctrodos na grelha, é necessário recorrer à posição guardada na estrutura `Electrode` para alinhar as vistas. Quando se usa a vista por grelha, os labels de cada um dos eléctrodos na vista 3D, são desactivados, com a excepção do eléctrodo sobre o qual foi feito o *click*.



### 3.3.11 Visualização de dados eléctricos

Para a visualização dos valores eléctricos sobre o cortex é necessário mapear as posições dos electrodos/grelhas onde foram medidos no cortex. Para representar a grelha é necessário criar então uma superfície a partir dos eléctrodos já colocados. Para isso foi usado o algoritmo de Delaunay 2D, definido através de uma triangulação que satisfaz o critério de Delaunay para simplexes de  $n$  dimensões ( neste caso  $n = 2$  e os simplexes são triângulos).

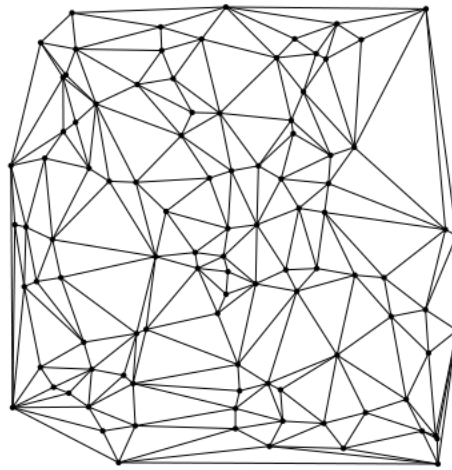


Figura 3.19: Exemplo de uma triangulação de Delaunay para 100 pontos

O critério declara que a circunferência de cada simplex numa triangulação contém somente  $n+1$  pontos do simplex. O VTK fornece a classe `vtkDelaunay2D` para executar este algoritmo a usando um conjunto de pontos (`vtkPoints`), e tem como output um `vtkPolyData` que vai servir para gerar a visualização da superfície.

Cada vertice do modelo definido, corresponde um eléctrodo, e consequentemente um valor para cada instante de tempo. É definido um array de valores escalares(`vtkFloatArray`) correspondentes à leitura para cada eléctrodo por instante.

Para a atribuição de uma escala de cores adequada, foi criada uma lookup table com 3 valores, (vermelho para o valor máximo de voltagem ( $200\mu V$ ), branco para 0, e azul para o mínimo ( $-200\mu V$ )).



### 3.3.12 Eléctrodos: propriedades

Cada eléctrodo pode ser catalogado com informação relativa tanto às funções das áreas adjacentes como à sua associação com a evolução da crise. Na estrutura *Electrode*, é guardado para cada eléctrodo, quais os labels que para ele estão activos. Para se poder utilizar, este mecanismo de classificação é necessário primeiro, seleccionar um ficheiro com o template da grelha de eléctrodos, só depois disto, se poderá navegar sincronamente as várias vistas, e classificar os eléctrodos. A partir deste ponto, ao utilizador basta navegar pelos eléctrodos, e mudar a meta-dados no interface próprio (ver Figura 3.22). Uma classe *callback* actualiza as estruturas de dados, cada vez que recebe um evento desse interface.



Figura 3.22: Interface de classificação de eléctrodos e sua visualização no template da grelha

### 3.3.13 Interface com o utilizador

O KWWidgets é construído sobre o Tcl/Tk e foi usado o seu interface em C++, mas é necessário inicializar o interpretador (Tcl é uma linguagem interpretada) antes qualquer outra coisa.

```
Tcl_Interp *interp = vtkKWApplication::InitializeTcl(argc, argv, &cerr)
```

Depois é criada a `vtkKWApplication`, que é a classe geral que representa toda a aplicação. Seguidamente uma `vtkKWWindow` que é adiciona à aplicação já criada. Para cada janela, pode-se associar uma variedade de *widgets* e dispo-los de várias formas. Cada um desses *widgets* pode gerar eventos, assim como o próprio VTK. Podemos escutar esses eventos e executar *callbacks* quando são despoletados. Os *callbacks* são classes, que pelo menos têm um método de execute, quando é referida a execução de um callback, é a esse método que se está a fazer referência.

Através desta secção é descrita a interface, enumeradas as classes usadas para a sua construção assim como o seu comportamento. A nomenclatura usada é a original, usada durante a programação do interface, e a sua tradução nem sempre é directa. Refere-se a vários tipos de elementos gráficos: botões, entrada de texto, janelas de *logging* e de *render*.

### 3.3.14 Inicialização

A inicialização começa com a instanciação da classe `Start`, e com a chamada dos métodos para criar todos os *widgets* necessários para a escolha do ficheiro, ou ficheiros, a ler.

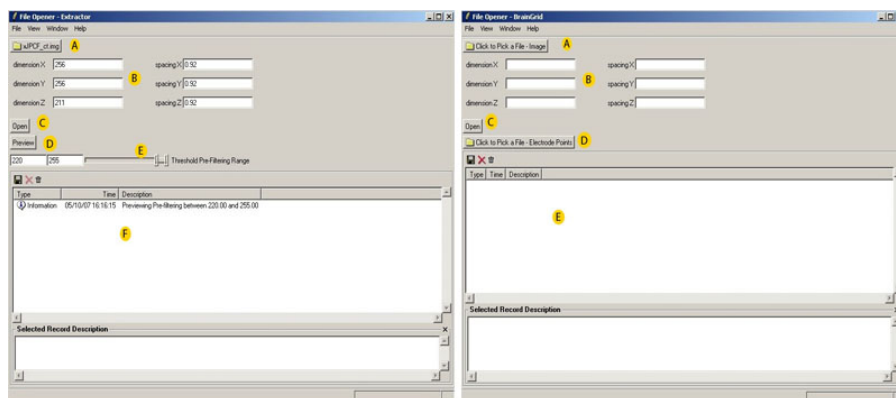


Figura 3.23: Interface Inicial

Na Figura 3.23 está o resultado de diferentes instanciações da classe `Start`. Para a construção da classe foram usadas as *widgets*: `vtkKWPushButton` (C e D), `vtkKWLoadSaveButton` (A à esquerda, A e D à direita), `vtkKWEntryWithLabel` (B), `vtkKWSplitFrame` (usado para organizar as caixas de texto), `vtkKWLogWidget` (F à esquerda, E à direita) e `vtkKWRange` (E). Apenas os seguintes geram eventos escutados por *callbacks* :

- Botão `vtkKWPushButton` (C) - quando clicado, é executado o *callback* `openCallback`, que é responsável pela criação de uma nova `vtkKWApplication` e instanciação das classes `Widgets`, `Extractor` e `BrainGrid`. É aqui que estas classes recebem os valores das dimensões da imagem e os seus paths.
- `vtkKWPushButton` (D) - quando clicado, é executado o *callback* `previewCallback`, que é responsável pela criação de uma `vtkKWApplication` e da instanciação das classes `Widgets` e `Preview`, esta ultima que gera a imagem de preview para a aplicação de extracção dos eléctrodos.
- `vtkKWLoadSaveButton`(A à esquerda, A e D à direita) - este *widget* abre uma caixa especial para selecção de um ficheiro, após seleccionado executa o *callback* `fileSelectorCallback`, que lê o cabeçalho da imagem médica e actualiza as caixas de texto responsáveis pelas dimensões e spacing.

### 3.3.15 Preview

Esta aplicação é responsável pelo preview do filtro `vtkImageThreshold` para a extracção de eléctrodos, lê a imagem como explicado na secção Leitura (3.3.2, página 34), e apresenta o resultado numa janela independente.

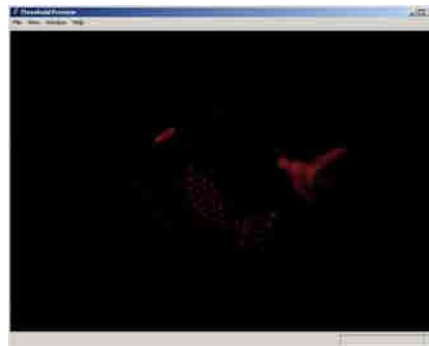


Figura 3.24: Janela do preview.

### 3.3.16 Co-Reg

Esta aplicação é responsável por correr o algoritmo de co-registo da imagem de TC com a RM. No interface, mostrado na figura 3.25, é possível seleccionar a imagem original e a que vai ser co-registada, assim como o ficheiro que vai conter o resultado.



Figura 3.25: Janela do Co-Reg e algoritmo de co-registo a correr.

## 3.3.17 Manual Extractor

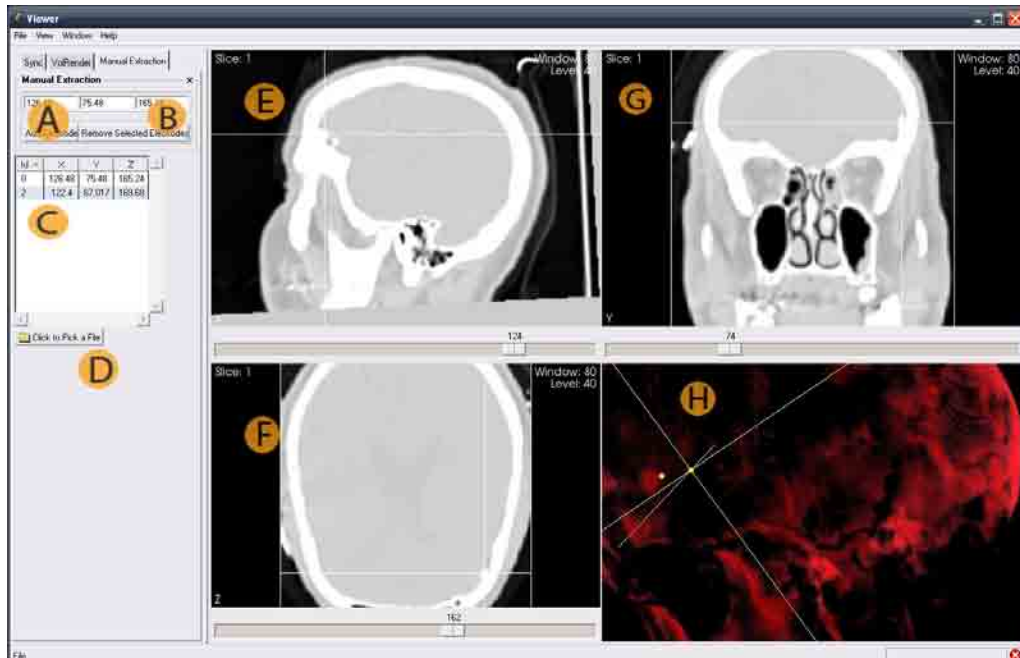


Figura 3.26: Interface de Extracção Manual de Eléctrodos

Aqui é mostrado o interface da aplicação de extracção manual de eléctrodos (Figura 3.26). Para a construção dos métodos relativos aos extractor foram usadas as widgets: `vtkKWButtonSet`(A e B), `vtkKWMultiColumnListWithScrollbars`(C), `vtkKWLoadSaveButton`(D). Os seguintes geram eventos escutados por *callbacks* :

- `vtkKWButton` (A) - quando clicado, executa o *callback* `addElectrodeCallback`, que adiciona o eléctrodo à lista, com o seu id e posição e também à visualização 3D.
- `vtkKWButton` (B) - quando clicado, executa o *callback* `removeElectrodeCallback`, que remove o eléctrodo seleccionado da lista e da visualização 3D.
- `vtkKWLoadSaveButton` (D) - quando clicado, executa o *callback* `extractCallback`, que escreve a lista de eléctrodos para um ficheiro, no formato correcto.

### 3.3.18 Extractor

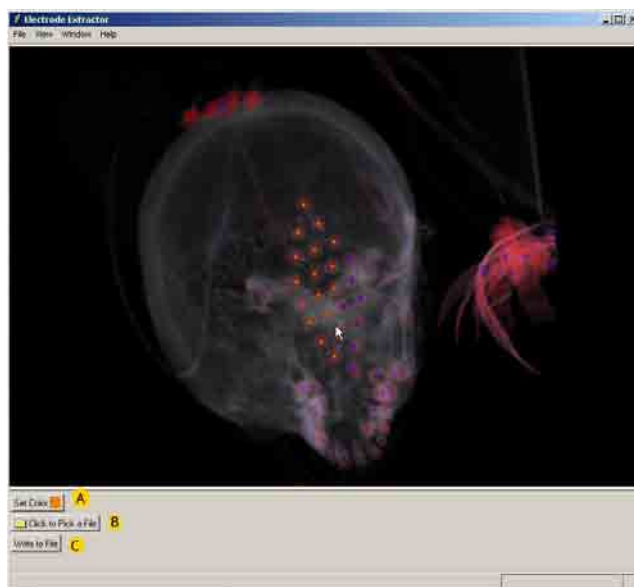


Figura 3.27: Interface de Extração de Eléctrodos

Nesta secção é ilustrado o resultado do algoritmo de extração de eléctrodos, e a selecção dos que são efectivamente eléctrodos (Figura 3.27). Para a construção dos métodos relativos aos extractor foram usadas as widgets: `vtkKWLoadSaveButton(B)`, `vtkKWChangeColorButton(A)`. Os seguintes geram eventos escutados por *callbacks* :

- `vtkKWPushButton (C)` - quando clicado, executa o *callback* `extractCallback`, que escreve para o ficheiro seleccionado as posições dos eléctrodos marcados.



### 3.3.19 BrainGrid

Nesta secção é mostrado a imagem do RMI, eléctrodos e electrocorticografia. Para a construção dos métodos relativos ao interface principal do BrainGrid foram usadas *widgets* seguintes (ver Figura 3.28).

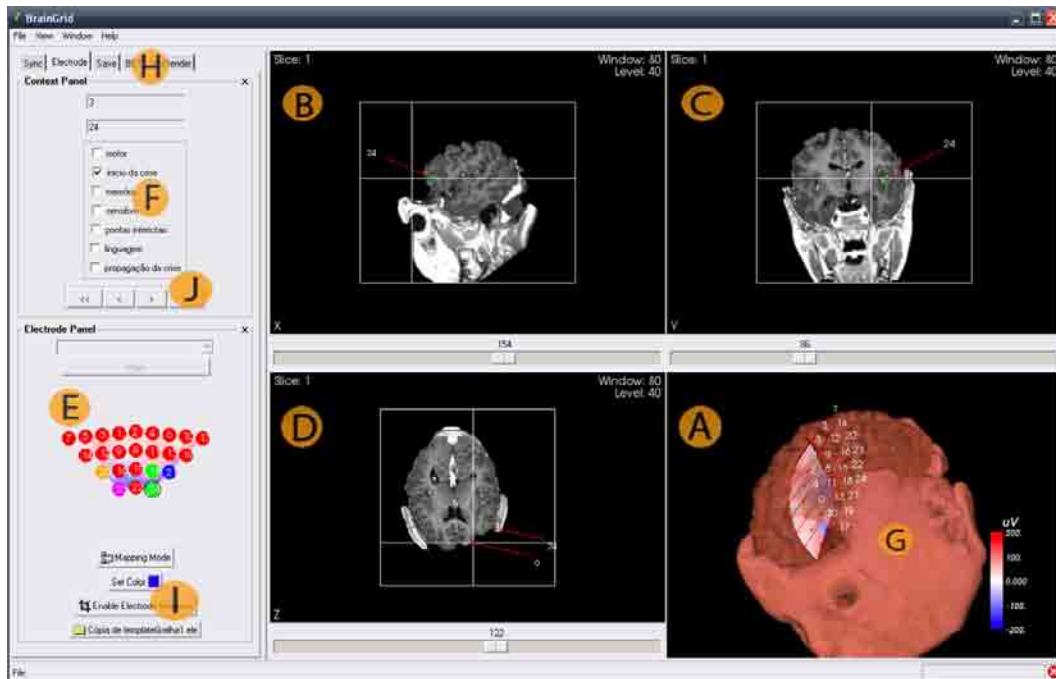


Figura 3.28: Interface Principal

- `vtkKWscale` (B, C, D em baixo) - quando movimentado, executa o *callback* `SliderCallback` respectivo, este é responsável pela actualização do viewer respectivo.
- `vtkKWRenderWindow` (A, B, C, D e E)
  - A - é janela de rendering principal, permite a manipulação nativa do VTK com o estilo de interacção em modo "joystick", permite com o rato fazer rotações, translações e zoom. Quando é feito um click sobre a janela é executado o *callback* `Sync` que por sua vez executa o *callback* `MainPicker`, Isto acontece para permitir fazer *picking* apenas com um click do rato. No caso de o click ter sido feito sobre um eléctrodo, ele selecciona-lo, e dispara as operações de sincronização para os viewers 2D, template de grelha de eléctrodos e painel de meta-dados.
  - B, C, D - são as janelas dos Viewers 2D, axial, coronal e sagital respectivamente. Por estar associada a um viewer, tem um estilo de interacção mais limitado, e permite alteração de contraste e brilho através da utilização do rato. Quando é feito um click sobre qualquer um deles, é executado o *callback* de sincronização (`SyncViewerX`, `Y`, `Z`) ou de *picking* (`PickerRenderX`, `Y`, `Z`) respectivo, consoante o estado modo de marcação ser activo ou inactivo.

- E - é a janela do template da grelha de eléctrodos, permite uma manipulação muito reduzida (`vtkInteractorStyleImage`). Quando é feito um click sobre a imagem, é executado o *callback* `GridSync`, que por sua vez pode executar o *callback* `GridPick`. Isto acontece novamente para ser possível fazer *picking* apenas com um click do rato. No caso do click ter sido feito sobre um eléctrodo da grelha, este é seleccionado, e são disparadas as operações de sincronização para os viewers 2D, viewer3D e painel de meta-dados.
- `vtkKWButtonSet` (J), estes são botões responsáveis pela navegação dos eléctrodos. O primeiro botão selecciona o primeiro eléctrodo, o ultimo botão selecciona o ultimo eléctrodo. Os botões intermédios seleccionam o eléctrodo anterior e o eléctrodo seguinte. Estão associados aos *callbacks* `NavigationCallback` (1, 2, 3 e 4), dependentes do `MainNavigationCallback` que trata de toda a sincronização dos dados.
- `vtkKWCheckBoxes` (F) - através destas caixas pode ser alterada a meta-dados, cada vez que é mudado o estado de alguma delas, é executado o *callback* `casosCheckButtonCallback`, que actualiza nas estruturas internas os dados.
- `vtkKWLoadSaveButton` (I) - este botão abre uma caixa para selecção de ficheiro, neste caso para a selecção do template da grelha de eléctrodos, depois de seleccionado o ficheiro é executado o *callback* `GridLoadCallback` que trata da leitura, e actualização dos meta-dados, assim como da colocação da grelha.
- `vtkKWPushButton` (E) - quando clicado, é executado o *callback* `ComboBoxGridCallback` responsável pela marcação entre os eléctrodos extraídos e os eléctrodos da grelha.
- `vtkKWComboBox`(E) - a caixa permite a selecção de um ID de um eléctrodo extraído para fazer corresponder um dos eléctrodos da grelha que esteja seleccionado. Funciona em conjunto com o botão `mark`, descrito anteriormente.
- e ainda `vtkKWCheckButton` (I), `vtkKWEntry` (H), `vtkKWChangeColorButton` (I), `vtkKW-Notebook` (H).

Existem ainda uma série de opções apresentadas na figura 3.29.

- `vtkKWButtonSet` (A), estes botões são responsáveis pela alteração do Zoom nas vistas 2D. Existem 3 presets, zoom uma vez, duas vezes e quatro vezes. Em alternativa à alteração do zoom usando o rato.
- `vtkKWButtonSet` (B), estes botões são responsáveis pelo activar e desactivar da numeração dos eléctrodos.
- `vtkKWButtonSet` (C), estes botões são responsáveis pelo activar e desactivar da imagem 3D da RM, por motivos de visualização ou de performance. E também da visualização dos dados eléctricos da ECoG.
- `vtkKWLoadSaveButton` (D), este botão é responsável pela gravação num ficheiro dos dados da marcação dos eléctrodos relativos a um template de eléctrodos específico.
- `vtkKWLoadSaveButton` (E), este botão é responsável pelo carregamento do ficheiro com as leituras do ECoG.



Figura 3.29: Side Bar - todas as opções não mostradas no interface principal

- `vtkKWButtonSet` (F), estes botões são responsáveis pelo lançamento do algoritmo de segmentação do cortex, o BET, é possível alterar a constante predefinida para melhores resultados.
- `vtkKWScaleWithEntry` (G), este *widget* é responsável pelo resampling ou não, da imagem 3D da RM, serve apenas para melhorar a performance quando necessário.
- O conjunto de *widgets* H, I e J, servem para calibrar o algoritmo de volume Rendering. É possível mudar o tipo de interpolação, entre linear e por aproximação. Mudar as funções transferencia relativas à cor e à opacidade, especificando os níveis, intensidades e ou cores. Tendo o histograma da imagem como guia. Finalmente o botão J serve para aplicar estas alterações, uma vez que não são automáticas.

## Capítulo 4

# Resultados

A aplicação BrainGrid desenvolvida permite visualizar dados médicos em formato Analyze, provenientes de RMIs e TCs, através de várias formas de visualização sincronizada. Permite a leitura de um TC e extracção semiautomática dos eléctrodos. Permite a sobreposição dessa informação posicional dos eléctrodos previamente extraídos automaticamente ou manualmente, bem como a associação a um template de grelha de eléctrodos. Permite também a leitura da informação eléctrica proveniente de uma ECoG, e sua visualização (como prova de conceito, os dados usados para a informação são os mesmos através dos vários exemplos). Finalmente é possível a associação de metadados a cada eléctrodo, estando cada tipo de dados associado a uma cor diferente. Os dados podem ser guardados, num ficheiro com um formato próprio.

Para ilustrar estes resultados, é mostrado um ciclo completo de utilização do programa usando a extracção automática e outro usando a extracção manual:

- Extracção de eléctrodos
  - Automática
  - Manual
- BrainGrid
  - Inicialização
  - Classificação
  - Visualização dos dados eléctricos

## 4.1 Extracção de eléctrodos

### 4.1.1 Automática

A extracção de eléctrodos é feita a partir da imagem de uma TC.

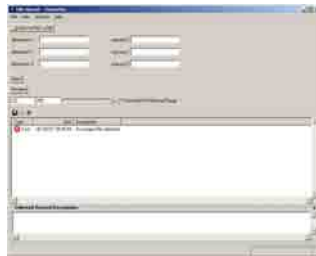


Figura 4.1: Extracção de eléctrodos - selecção da imagem

O utilizador faz a selecção da imagem a abrir, e confere as dimensões e espaçamento. Pode ser feito um ajuste dos níveis de cor a usar no algoritmo de extracção, e *preview* da filtragem resultante desse intervalo. Como mostrado na Figura 3.24, página 51.

O algoritmo é computacionalmente intensivo e não consegue detectar os eléctrodos com precisão em situações de sobreposição. O resultados do algoritmo vão sendo mostrados na janela de *logging*.

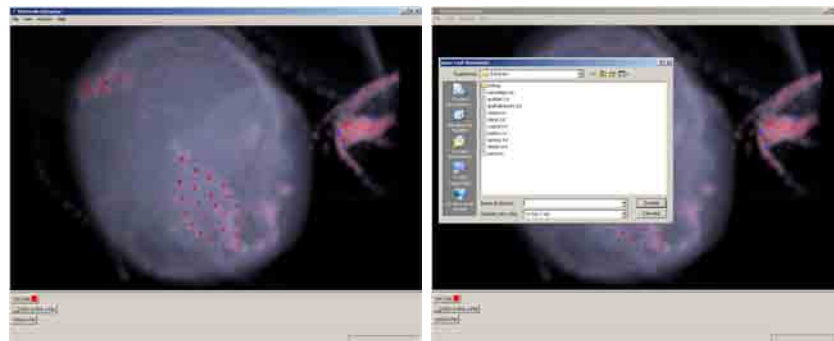


Figura 4.2: Extracção de eléctrodos - sequência da selecção de eléctrodos.

Após a conclusão do algoritmo são mostrados no ecrã os possíveis eléctrodos juntamente com a imagem original do TC. O utilizador selecciona cada um dos eléctrodos reais e pode seleccionar um ficheiro para guardar o resultado, como mostrado na sequência da Figura 4.2. Este algoritmo é ainda bastante rudimentar, e tem que ser considerado apenas um ponto de partida. Dos 7 exemplos testados, só foi possível usar o algoritmo em 3 deles (ver anexo C página 82).

### 4.1.2 Manual

A extracção pode também ser feita manualmente, usando o extractor manual. Desta forma é eliminado o problema de eléctrodos sobrepostos, e de pontos confundidos com eléctrodos. É muito mais flexível e foi possível testar todos os exemplos. Mas requer algum treino por parte do utilizador.

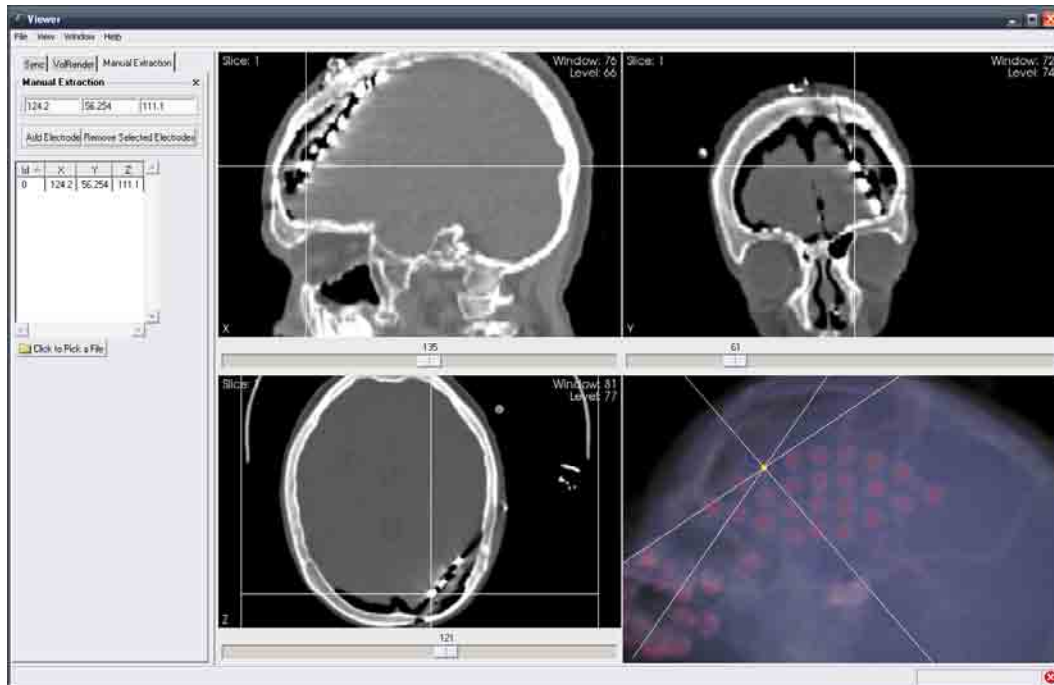


Figura 4.3: Extracção manual de eléctrodos - parte 1



Figura 4.4: Extracção manual de eléctrodos - parte 2 e 3

No extractor manual o utilizador faz também a selecção da imagem a abrir, e confere as dimensões e espaçamento. Aqui no entanto tem que seleccionar, clicando na vista 2D, as zonas correspondentes aos eléctrodos, e afinar a sua posição usando os *sliders*, e confirmando para cada eléctrodo carregando no botão para adicionar o ponto à lista de eléctrodos e à vista 3D para

melhor ilustrar a posição relativa deste. Pode a qualquer momento retirar qualquer ponto da lista. Quando concluir, pode gravar a lista de eléctrodos usando a opção de gravação (ver Figuras 4.3 e 4.4).

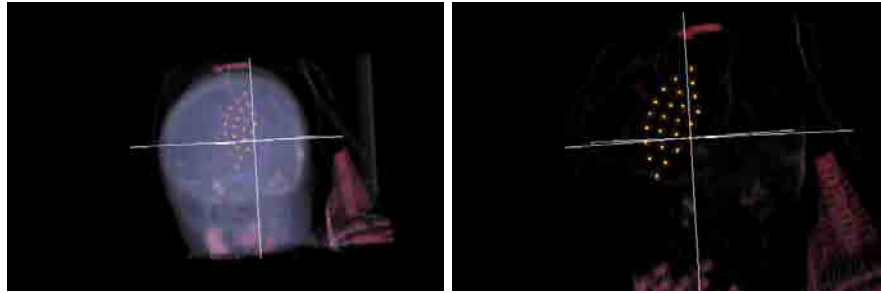


Figura 4.5: Vista 3D, alteração de opacidades para melhor visualização dos eléctrodos

De notar, que foi incluído no extractor manual a opção de afinar a vista 3D. Esta opção permite ao utilizador, através de uma serie de *widgets* específicos, melhorar a visualização 3D com o objectivo de melhor extracção dos eléctrodos (ver Figura 4.5).

## 4.2 BrainGrid

Esta é a aplicação para visualização do RMI e dados de ECoG. O utilizador faz a selecção da imagem a abrir, e confere as dimensões e espaçamento. Selecciona também o ficheiro contendo as posições dos eléctrodos extraídos.

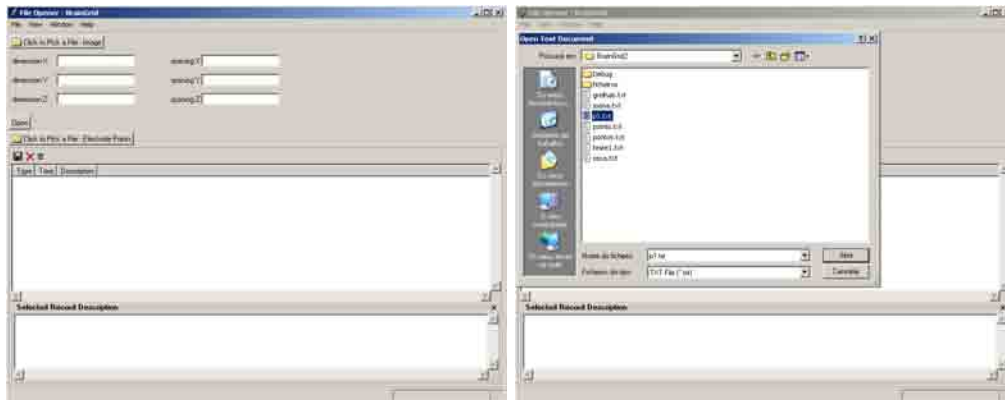


Figura 4.6: Sequência de selecção dos dados a visualizar

Seguidamente é possível fazer a sua visualização multimodal, isto é, temos a vista 3D da imagem RMI com os eléctrodos e suas labels. As vistas 2D axial, coronal e sagital, também com os eléctrodos (por defeito a cor de um eléctrodo sem meta-dados associados é vermelho). O utilizador pode alterar o zoom das imagens usando os botões no painel (zoom 1x, 2x e 4x) ou usando a wheel do rato (ver Figura 4.7). Também usando o rato, é possível fazer rotações e translações da imagem 3D. Pode também activar e desactivar a imagem 3D do RMI, esta opção é especialmente vantajosa quando a imagem tem um resolução muito grande, podendo tornar muito pesada a sua manipulação. Activar e desactivar os labels dos eléctrodos e ainda da a visualização da grelha de eléctrodos (secção 3.3.11, página 46).

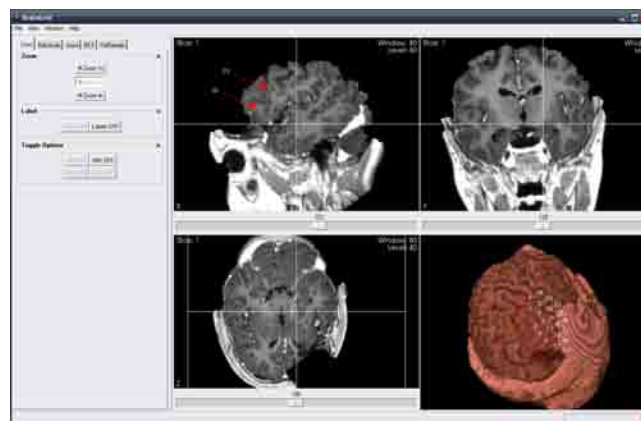


Figura 4.7: Visualização 3D e 2D com zoom 2x.



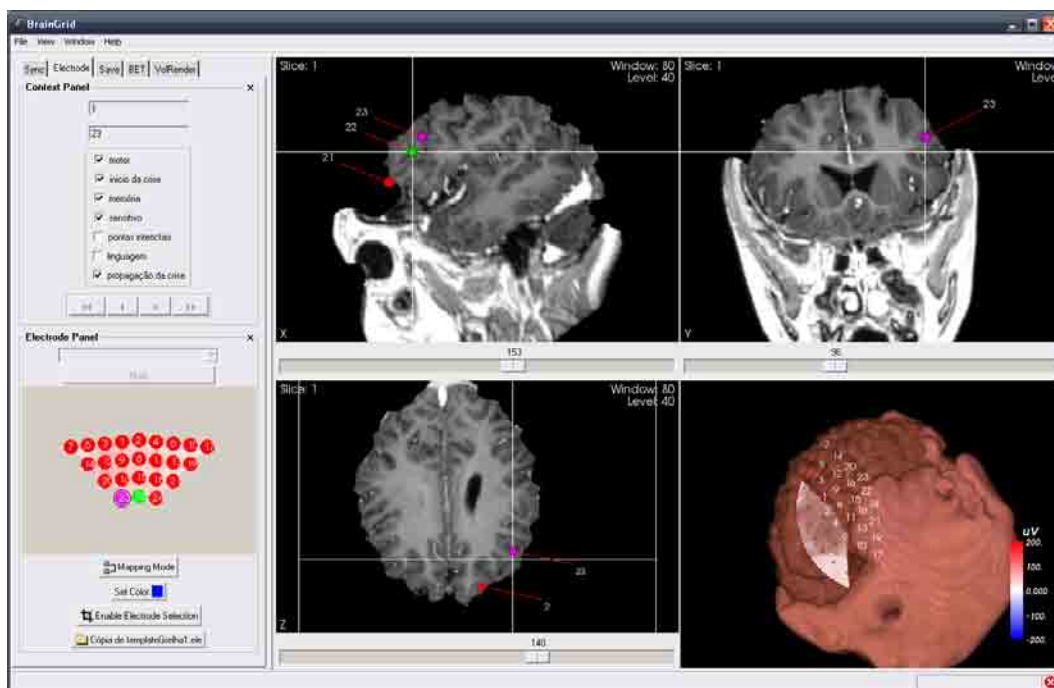


Figura 4.8: Inclusão da grelha de eléctrodos e alinhamento de vistas 2D com uso da mira e visualização da ECoG.

Mudando a tab de *Sync* para *Electrode* o utilizador revela um painel com novas opções para a navegação dos meta-dados e visualização do template da grelha de eléctrodos.

Fazendo o click sobre o botão para carregar um ficheiro no sub-painel *Electrode* (Figura 4.8), o utilizador pode seleccionar o template desejado. Estes templates variam, assim como as grelhas usadas no ECoG, alguns destes podem ser vistos no anexo B (página 78).

Este pode conter apenas as informações relativas ao template ou meta-dados associados a cada eléctrodo. A aplicação apresenta o template da grelha de eléctrodos, na qual o utilizador pode seleccionar qualquer eléctrodo que a aplicação encarrega-se de sincronizar as várias vistas de modo a que ele fique visível em todas elas. Na vista 3D o eléctrodo seleccionado é o unico com label de forma a distingui-lo mais facilmente.

Para uma visualização mais eficaz e precisa, na visualização 2D, é possível fazer um click num ponto da imagem, e a aplicação alinha automaticamente as outras, sendo usadas miras para relacionar as posições (Figura 4.8).

Os tipos de meta-dados são visíveis no painel dentro do sub-painel *Context*, os tipos predefinidos são:

- motor
- memória
- linguagem
- sensitivo
- pontas interictais
- propagação da crise

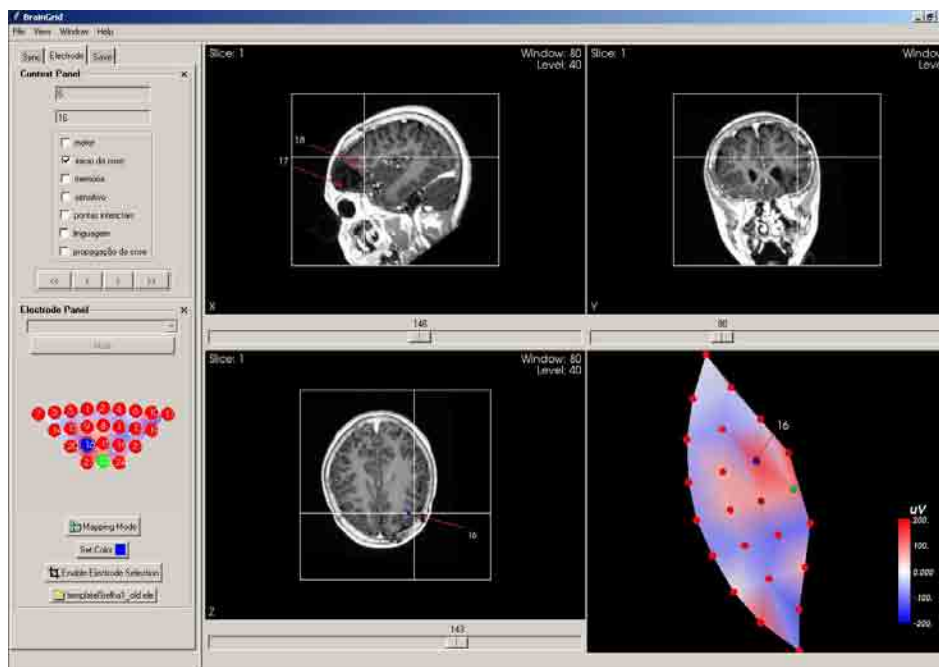


Figura 4.9: Alteração dos meta-dados (em cima), e gravação do estado da marcação (em baixo).

Como se pode ver na Figura 4.8 o eléctrodo 22 está marcado como estando colocado sobre a zona motora. Cada tipo de meta-dados está associado a uma cor diferente de forma a facilitar a distinção.

Para efectuar a visualização dos dados eléctricos é necessário já ter mapeado a grelha de eléctrodos com os eléctrodos extraídos (neste caso já estão mapeados). O utilizador escolhe o ficheiro de ECoG desejado, e a aplicação mostra na vista 3D e na vista do template da grelha de eléctrodos a informação das diferenças de potencial através de uma escala de cor, também mostrada para referência. Para navegar através da ECoG, estão disponíveis as teclas:

- '+' - permite avançar no tempo.
- '-' - permite recuar no tempo.

Imaginando que o utilizador, navega a ECoG até ao instante mostrado na Figura 4.9, e nota uma diferença de potencial muito elevada, coincidente com o eléctrodo 16. Este vai seleccioná-lo e marca-lo como sendo o início da crise. Para o seleccionar pode fazer um click sobre o eléctrodo na vista 3D ou na vista de grelha, e para o marcar, escolhe no sub-painel Context o meta-dado relativo ao início da crise. A aplicação mostra o eléctrodo com a cor azul nas várias vistas e actualiza os meta-dados.

O utilizador pode a qualquer momento guardar o estado das marcações no ficheiro do template da grelha de eléctrodos na tab Save do painel de ferramentas.

Neste exemplo, foi usada uma imagem de RMI do cortex. Caso a imagem contivesse toda a cabeça do paciente, poderia ser usado o Brain Extraction Tool (ver F na figura 3.29 na página 56), que permite obter um efeito igual ao do exemplo revisto.

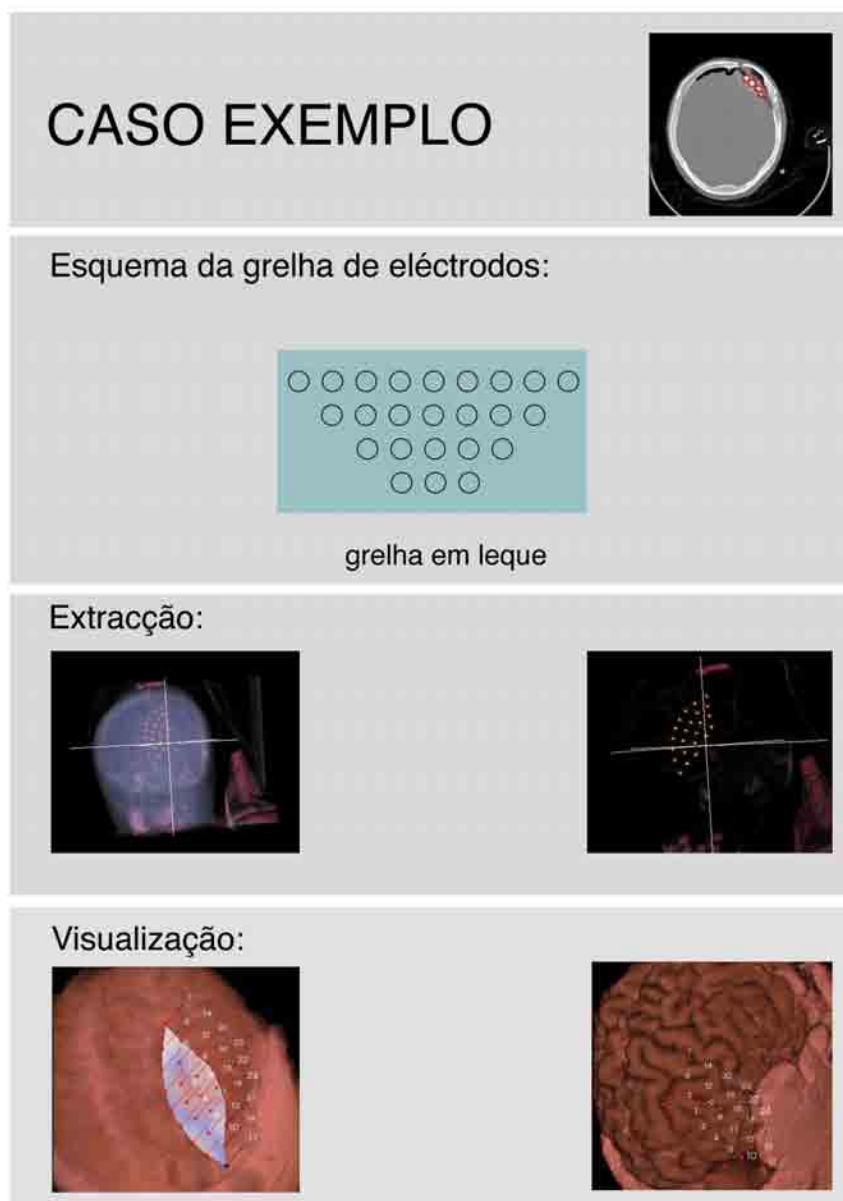


Figura 4.10: Exemplo

Na Figura 4.10 pode ver-se um resumo do processo para este caso. No anexo C (página 82). pode ver os outros resultados

## Capítulo 5

# Conclusão

Como prova de conceito, construímos uma aplicação que suporta todo o processo requerido para a visualização ECoG - RMI. Processo este por nós definido: co-registo TC/RMI, Brain Extraction Tool, extracção de eléctrodos manual e automática, mapeamento dos eléctrodos com as medições do ECoG, visualização combinada dos dados 3D e 2D de múltiplas fontes de informação (RMI, TC e ECoG) num interface simples como foi inicialmente planeado.

Como maior contribuição para o desenvolvimento futuro, apresenta-se a prova de conceito do processo necessário para uma visualização eficaz da ECoG. É também de realçar, o grande esforço em encontrar uma forma eficaz de visualizar a informação da morfologia do cérebro em 3D. Foram feitos vários testes e foi adoptado um método que privilegia o detalhe da informação. De forma a ser flexível foi construído um interface que permitisse tanto a continuação de testes como a adaptação da visualização a qualquer tipo de imagem.

Foram sentidas diversas dificuldades, especialmente ao nível da utilização do KWWidgets, já que esta é uma biblioteca bastante recente, ainda em crescimento e evolução, com muito pouca documentação e ajuda quase inexistente na web. Houve também uma tentativa de fazer um desenvolvimento paralelo Windows/Linux para que a aplicação fosse suportada nas duas plataformas, mas mais uma vez devido à falta de documentação esta não foi concretizada, optou-se assim por adoptar apenas a plataforma Windows para uma melhor rentabilização do tempo.

Para que esta ferramenta seja eficaz no meio clínico existem varios caminhos possíveis de percorrer. Estes caminhos passam pela melhoria na extracção de eléctrodos, a adopção de outros formatos standard, a melhoria da robustez e usabilidade do interface e maior flexibilidade na visualização 3D para que seja exigida uma menor intervenção do utilizador para um resultado ideal.

No futuro, seria relevante a criação de um formato para a representação das grelhas, associação da informação da meta-data e mapeamento dos eléctrodos com o ECoG. Possivelmente adoptando XML como solução de forma a facilitar a leitura através futuras aplicações.

O algoritmo actual de extracção de eléctrodos do ECoG necessita ainda de melhoramentos para reduzir o esforço computacional e melhorar a precisão da detecção. Há ainda que lidar com o problema do uso de várias grelhas de eléctrodos em simultâneo, situação que ocorre frequentemente em casos mais complexos.

Outro ponto interessante a rever seria permitir a marcação de zonas de interesse, além da associação de uma etiqueta a um eléctrodo, que permitisse demarcar as zonas da fala, movimento ou relativas à crise.

É possível no futuro a integração de outros métodos implementados no Insight Segmentation and Registration Toolkit (ITK) [itk, 2008], para melhorar o algoritmo de extracção e introduzir outras ferramentas e filtros na nossa aplicação.

As evoluções futuras deverão ser planeadas com os utilizadores finais desta aplicação de forma a afiná-la e adaptá-la a um ambiente clínico realista.

# Bibliografia

- [ana, 1990] (1990). Analyze, analyze 7.5 file format, <http://www.mayo.edu/bir/PDF/ANALYZE75.pdf>.
- [jul, 2007] (2007). Julius, june 2007 [http://www.julius.caesar.de/index.php/Julius\\_MV1](http://www.julius.caesar.de/index.php/Julius_MV1).
- [sli, 2008] (2008). 3d slicer, <http://www.slicer.org/>.
- [3DV, 2008] (2008). 3dviewnix <http://www.mipg.upenn.edu/Vnews/index.html>.
- [ami, 2008] (2008). Amide: e amide's a medical imaging data examiner, <http://amide.sourceforge.net/>.
- [bra, 2008] (2008). Brainsuite 2 <http://brainsuite.usc.edu/>.
- [dic, 2008] (2008). Dicom, nema's official dicom webpage, <http://medical.nema.org/dicom/2004.html>.
- [etd, 2008] (2008). etdips - exploratory two/three dimensional image processing system, <http://clinicalcenter.nih.gov/cip/software/etdips/>.
- [ezd, 2008] (2008). ezdicom <http://www.sph.sc.edu/comd/rorden/ezdicom.html>.
- [fsl, 2008] (2008). Fmrib software library <http://www.fmrib.ox.ac.uk/fsl/>.
- [fox, 2008] (2008). Fox, <http://www.fox-toolkit.org/>.
- [itk, 2008] (2008). Insight segmentation and registration toolkit, itk, may 2008 <http://www.itk.org/>.
- [kit, 2008] (2008). Kitware, <http://www.kitware.com/>.
- [kww, 2008] (2008). Kwwidgets, <http://www.kwwidgets.org/>.
- [mri, 2008] (2008). Mricro, <http://www.sph.sc.edu/comd/rorden/mricro.html>.
- [nam, 2008] (2008). Namic (national alliance for medical image computing), <http://www.namic.org/>.
- [nif, 2008] (2008). Neuroimaging informatics technology initiative (nifti), may 2008 <http://nifti.nimh.nih.gov/>.
- [ope, 2008] (2008). Opengl, <http://www.opengl.org/>.
- [qt, 2008] (2008). Qt, <http://trolltech.com/products/qt>.

- [spm, 2008] (2008). Spm (statistical parametric mapping) <http://www.fil.ion.ucl.ac.uk/spm/>.
- [vis, 2008] (2008). Vispack <http://www.cs.utah.edu/~whitaker/vispack/>.
- [vol, 2008] (2008). Volview, <http://www.kitware.com/products/volview.html>.
- [wxw, 2008] (2008). wxwidgets, <http://www.wxwidgets.org/>.
- [Cascino, 2002] Cascino, G. (2002). *Video-EEG monitoring in adults. Epilepsia* 43, pages 80–93.
- [Crawford, 2000] Crawford, P. (2000). *P. Epidemiology of intractable focal epilepsy. In: Oxbury, J. Polkey, C., Duchowny, M. Eds, Intractable Focal Epileps*, pages 25–40. Harcourt London.
- [Duncan, 1997] Duncan, J. (1997). *Imaging and epilepsy. Brain* 120, pages 339–377.
- [Duncan, 2002] Duncan, J. (2002). *Neuroimaging methods to evaluate the etiology and consequences of epilepsy. Epilepsy Res* 50, pages 131–140.
- [Engel, 1993] Engel, J. (1993). *Surgical treatment of the epilepsies*. Raven, New York.
- [Enger and Burr, 1994] Enger and Burr (1994). *Subdural electrodes. Acta Neurol Scand Suppl* 152.
- [Gastaut, 1973] Gastaut, H. (1973). *Dictionary of Epilepsy: Definitions World Health Organization*. Geneva.
- [Grant, 2004] Grant, P. (2004). *Structural MR imaging. Epilepsia* 45, pages 4–16.
- [Hauser, 1998] Hauser, A. (1998). *Incidence and prevalence. In: Engel, J., Pedley, T. Eds, Epilepsy: a Comprehensive Textbook*. Lippincott Raven, Philadelphia, PA.
- [Jayakar, 1999] Jayakar, P. (1999). *Invasive EEG monitoring in children: when, where, and what? J Clin Neuro-physiol* 16, pages 408–418.
- [Knowlton, 2004] Knowlton, R. (2004). *Multimodality imaging in partial epilepsies. Curr Opin Neurol* 17, pages 165–172.
- [Koepp, 2005] Koepp, M.J., W. F. (2005). *Imaging structure and function in refractory focal epilepsy. Lancet Neurol* 4, pages 42–53.
- [Luders, 1992] Luders, H., A. I. (1992). *Conceptual considerations. In: Luders, H. Ed Epilepsy Surgery*, pages 51–62. Raven, New York.
- [Malmivuo, 1997] Malmivuo, J., S. V. E. H. (1997). *Sensitivity distributions of EEG and MEG measurements. IEEE Trans Biomed Eng* 44, pages 196–208.
- [O'Brien, 1999] O'Brien, T.J., S. E. M. B. H. M. B. B. J. C. J. C. G. M. F. S. F. (1999). *Subtraction SPECT co-registered to MRI improves postictal SPECT localization of seizure foci. Neurology* 52, pages 137–146.
- [Richardson, 2003] Richardson, M. (2003). *Epilepsy and surgical mapping. Br Med Bull* 65, pages 179–192.

- [Rosenow, 2001] Rosenow, F., L. H. (2001). *Presurgical evaluation of epilepsy. Brain* 124, pages 1683–1700.
- [Smith, 2002] Smith, S. (2002). *BET: Brain Extraction Tool, 200x, Stephen M. Smith , FMRIB technical Report TR00SMS2b.*
- [W. J. Schroeder, 1998] W. J. Schroeder, K. M. M. (1998). *The Visualization Toolkit An Object Oriented Approach to 3D Graphics.* Prentice Hall.
- [W. J. Schroeder, 2001] W. J. Schroeder, K. M. Martin, L. S. A. (2001). *The Visualization Toolkit User's Guide.* Prentice Hall.
- [Wiebe, 2001] Wiebe, S., B. W. G. J. E.-a. (2001). *A randomized, controlled trial of surgery for temporallobe epilepsy, N Engl J Med.* 345(5): 311-8.



# Apêndice A

## - Arquitectura

### Pacote Visualização

O pacote Visualização (ver Figura A.1) inclui duas das classes principais. A classe Extractor, responsável na aplicação de extracção de eléctrodos pela visualização dos eléctrodos extraídos, e pelo algoritmo de extracção.

A classe BrainGrid, talvez a mais importante da aplicação, é a responsável pela leitura dos dados raw das imagens (secção 3.3.2, página 34) , e consequente visualização, esta classe é também responsável pela instanciação de alguns das classes *callbackde picking* e da classe PlaneWidgetControl, responsável pelo corte da imagem para consequente visualização 2D.

O pipeline de visualização da imagem 3D em volume rendering é da responsabilidade da classe VolRender (secção 3.3.3, página 34).

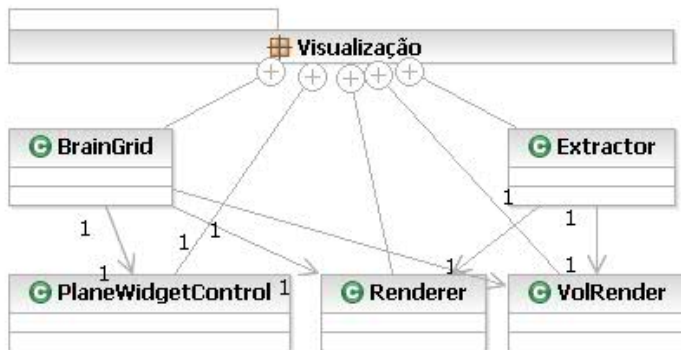


Figura A.1: Pacote Visualização

### Pacote Dados

O pacote Dados (ver Figura A.2) contém as estruturas que vão guardar os dados.

- Electrode - Armazena os dados relativos ao eléctrodo: ID, posição, cor, meta-dados, ID do eléctrodo na grelha de eléctrodos (se previamente mapeada) e ainda ponteiros para as suas representações na visualização 3D. Esta estrutura será usada num vector.
- Grid - Armazena os dados relativos ao template de grelha de eléctrodos: ID, posição, ID do eléctrodo extraído e ainda ponteiros para as suas representações na visualização do template. Esta estrutura será usada num vector.
- Current e CurrentGrid - são estruturas auxiliares na interacção e sincronização, guarda respectivamente os dados relativos ao eléctrodo seleccionado no template da grelha de eléctrodos e aos eléctrodos estão a ser mostrados em casa uma das vistas 2D.

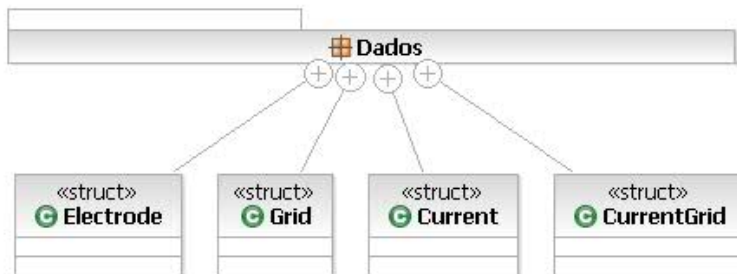


Figura A.2: Pacote Dados

### Pacote Load/Save

O pacote Load/Save (ver Figura A.3) contém as classes responsáveis pela selecção de ficheiros e carregamento da informação para o sistema ou a sua gravação.

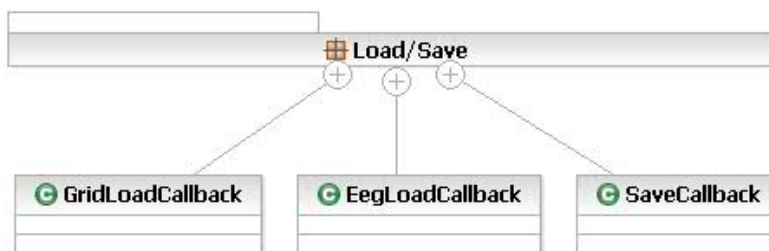


Figura A.3: Pacote Load/Save

- GridLoadCallback - Classe responsável pelo carregamento de toda a informação relativa ao template da grelha de eléctrodos.
- EegLoadCallback - Classe responsável pelo carregamento da informação relativa aos dados da eléctricos da electrocorticografia.
- SaveCallback - Classe responsável por gravar a meta-dados num ficheiro, formato do template da grelha de eléctrodos.

### Pacote ECoG

O pacote ECoG (ver Figura A.4) contém classes responsáveis pela leitura da informação eléctrica e pela criação do modelo na qual esta vai ser representada. Explicação detalhada na secção 3.3.11, página 46.

- EegSession - Classe contentora responsável pela leitura dos ficheiros .eeg. Agrega um conjunto de amostras (EegSample).
- EegSample - Classe que contém um vector de valores diferença de potencial (um para cada eléctrodo), bem como informação relativa ao índice do sample, e ao instante de tempo em que foi efectuado a amostragem.
- EegCallback - Classe responsável pela interacção para mudança do instante de tempo.
- GridModel - Classe responsável pela criação do modelo no qual a informação eléctrica vai ser representada.

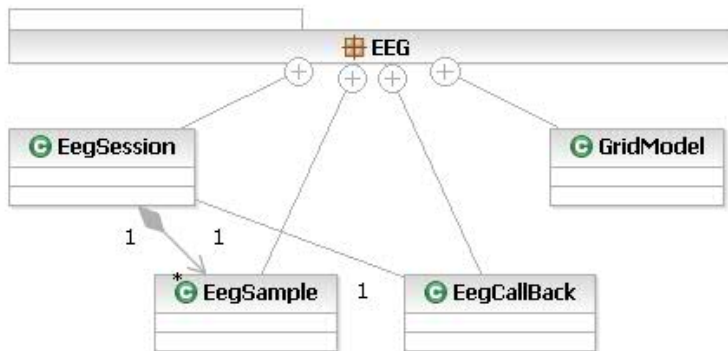


Figura A.4: Pacote ECoG

## Pacote Inicialização

O pacote Inicialização (ver Figura A.5) contém classes responsáveis pela inicialização da aplicação, selecção de ficheiros, correcção de valores de dimensões e espaçamento. É a classe que instancia a classe BrainGrid e Widget, para a inicialização da aplicação.

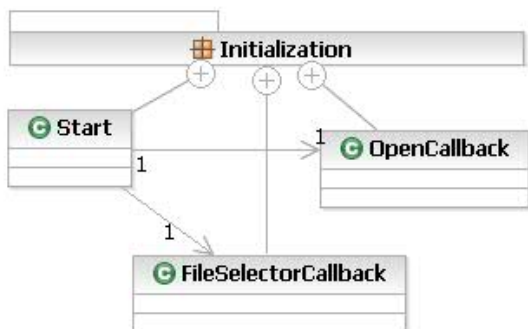


Figura A.5: Pacote Inicialização

## Pacote Tratamento de Dados

O pacote Tratamento de Dados (ver Figura A.6) contém classes responsáveis por correr os binários relativos aos algoritmos de co-registo e segmentação do cérebro.

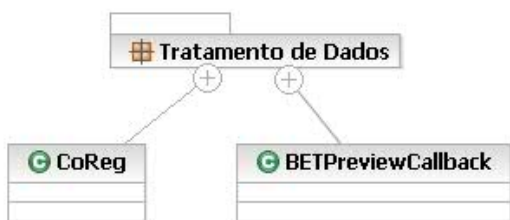


Figura A.6: Pacote Tratamento de Dados

## Pacote GUI

O pacote GUI (ver Figura A.7) contém as classes responsáveis pela criação do interface gráfico: desde as janelas VTK, botões, sliders, checkboxes, checkbuttons, comboboxes e outros. E também a instanciação das classes que definem o comportamento destes widgets. Apesar de ser só uma classe, é bastante extensa, e poderia e deveria ser dividida noutras. A descrição do comportamento de alguns dos *callback* instanciados por ela é descrita mais adiante.

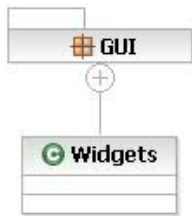


Figura A.7: Pacote GUI

## Pacote Auxiliar

O pacote Auxiliar (ver Figura A.8) contém classes que não se enquadravam em nenhum pacote especial, em comum têm o facto de serem usadas para controlar certos *widgets* do GUI.

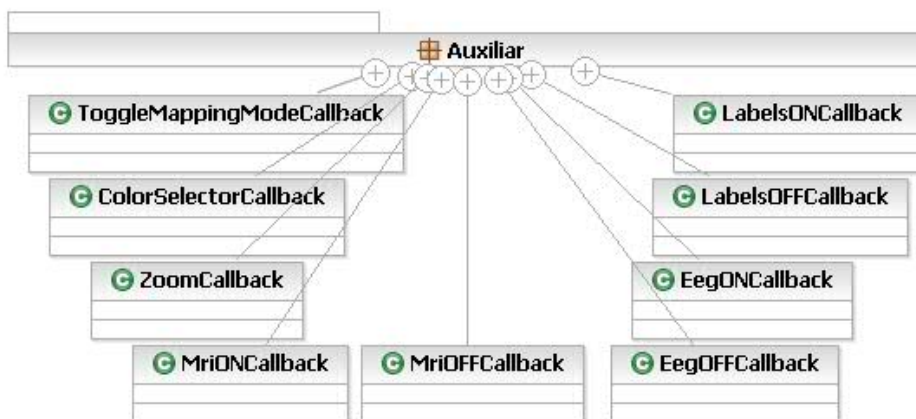


Figura A.8: Pacote Auxiliar

- ColorSelectorCallback - Classe *callback* responsável pela mudança da cor actual para marcação de eléctrodos.

- ZoomCallback - Classe *callback* responsável pela mudança do nível de zoom.
- ToggleMappingModeCallback - Classe responsável pela mudança do flag para mudança para o modo de mapeamento de eléctrodos.
- LabelsONCallback e LabelsOFFCallback - Classes responsáveis pela activação e desactivação das labels dos eléctrodos.
- eegONCallback e eegOFFCallback - Classes responsáveis pela activação e desactivação da grelha para mostrar o feedback visual da ECoG.
- mriONCallback e mriOFFCallback - Classes responsáveis pela activação e desactivação do volume do RMI.

### Pacote Navegação de Meta-Dados

O pacote Navegação de Meta-Dados (ver Figura A.9) contém as classes responsáveis pela navegação e actualização dos meta-dados. Explicação detalhada na secção 3.3.12, página 48.

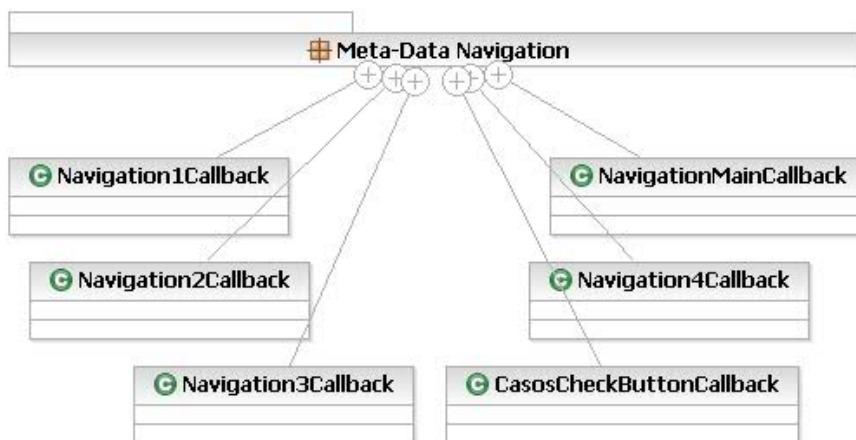


Figura A.9: Pacote Navegação de Meta-Dados

- Navigation1Callback e Navigation4Callback - Classes *callback* responsáveis para a selecção do primeiro ou ultimo eléctrodo da lista respectivamente.
- Navigation2Callback e Navigation3Callback - Classes *callback* responsáveis para a selecção do eléctrodo anterior ou seguinte respectivamente.
- NavigationMainCallback - Classe *callback* responsável pelo highlight do eléctrodo seleccionado e actualização do display da informação.
- CasosCheckButtonCallback - Classe *callback* responsável pela escrita para a estrutura de dados da meta-dados do eléctrodo seleccionado.

### Pacote Sliders

O pacote *Sliders* (ver Figura A.10) contém as classes responsáveis pelo controlo dos sliders criados, para navegar as vistas 2D axial (*SliderCallbackX*), coronal (*SliderCallbackY*) e sagital (*SliderCallbackZ*).

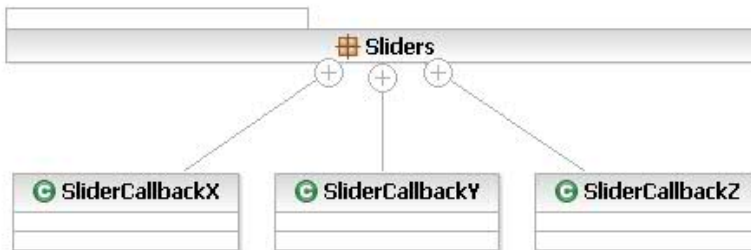


Figura A.10: Pacote Sliders

### Pacote Sincronização

O pacote Sincronização (ver Figura A.11) contém as classes responsáveis pela sincronização dos viewers 2D cada vez que ocorre um click, alinhamento das miras e do alinhamento dos planos de corte. O mecanismo de sincronização é explicado em detalhe na secção 3.3.10, página 45.

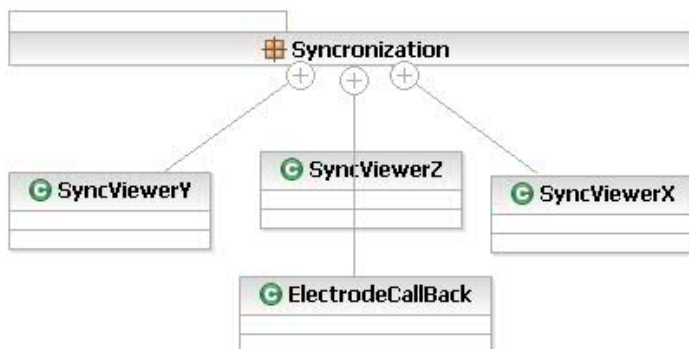


Figura A.11: Pacote Sincronização

### Pacote Pickers

O pacote *Pickers* (ver Figura A.12) contém as classes responsáveis pelo *picking* (selecção de um ponto ou eléctrodo).

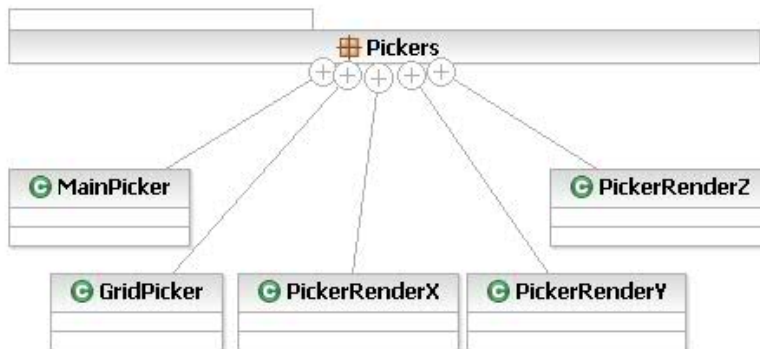


Figura A.12: Pacote Pickers

- *PickerRenderX*, *PickerRenderY*, *PickerRenderZ* - Classes *callback*responsáveis pela selecção de um ponto nos viewers 2D.
- *MainPicker* - Classe *callback*responsável pela selecção de eléctrodos na vista 3D.
- *GridPicker* - Classe *callback*responsável pela selecção de eléctrodos na vista de template de grelha de eléctrodos.



## Apêndice B

### - Templates de grelhas de eléctrodos

Nas figuras é mostrado o template da grelha de eléctrodos (em cima), a visualização da sua colocação sob a superfície do cortex (no meio) e o processo de extracção manual dessa mesma grelha (em baixo).

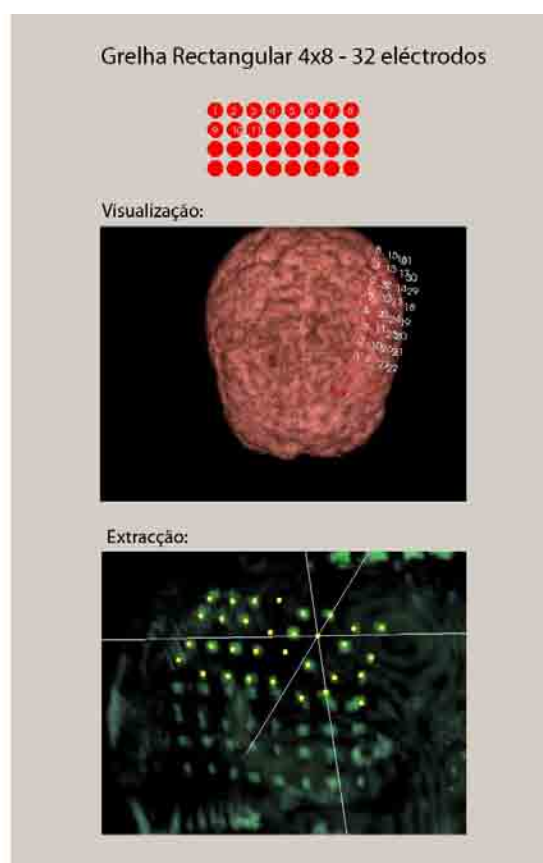


Figura B.1: Grelha rectangular 4x8 - 32 eléctrodos

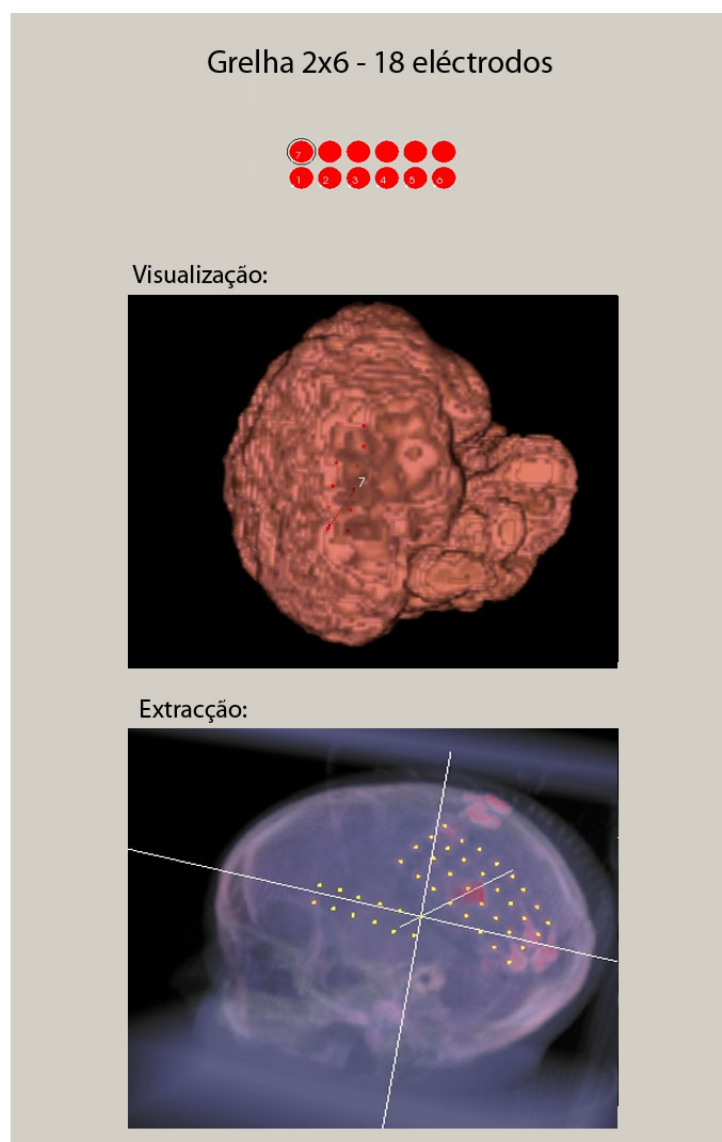


Figura B.2: Grelha rectangular 2x6 - 18 eléctrodos

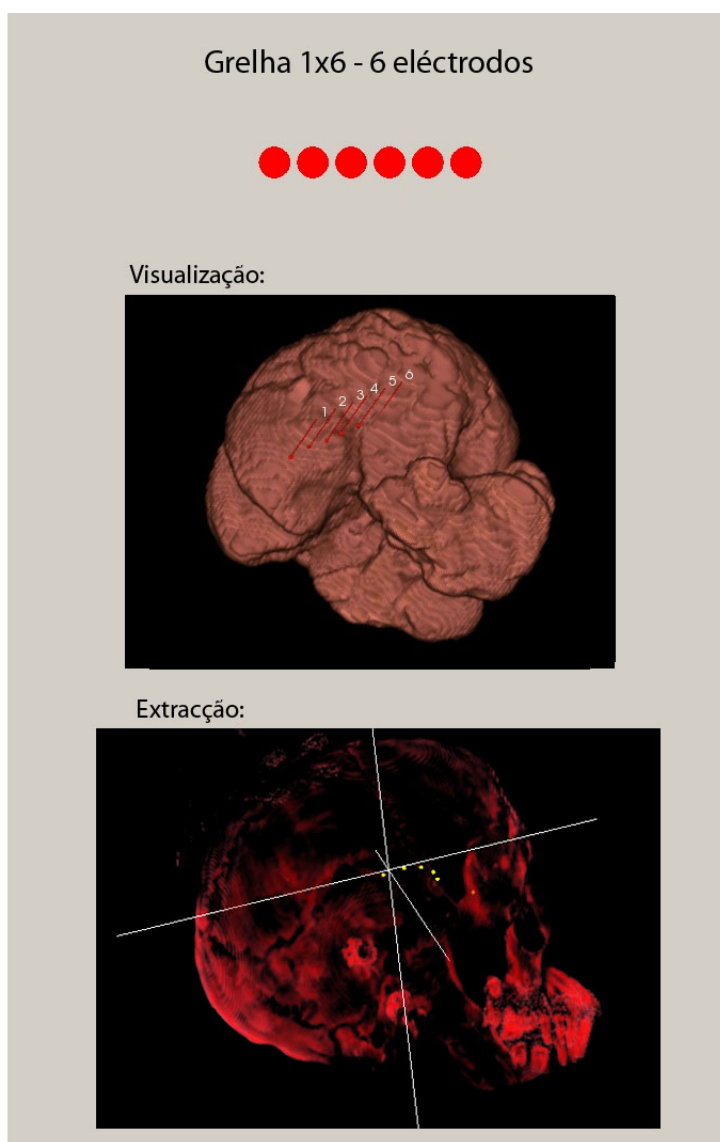


Figura B.3: Grelha em fita 1x6 - 6 eléctrodos

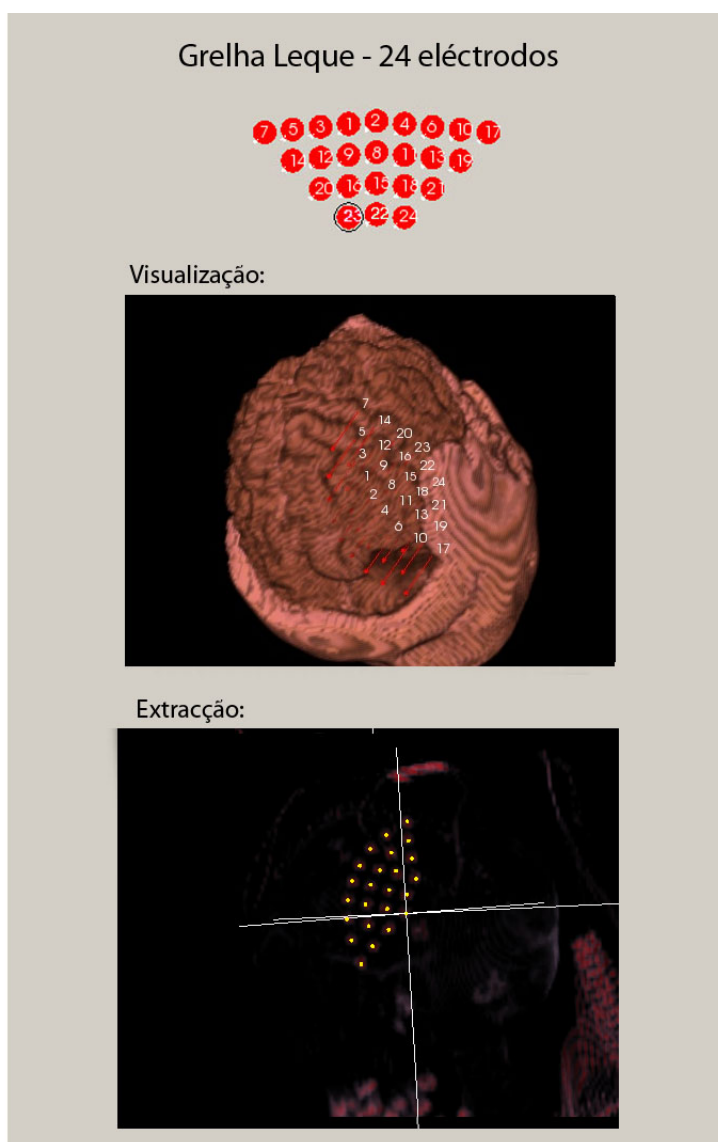


Figura B.4: Grelha em leque - 24 eléctrodos

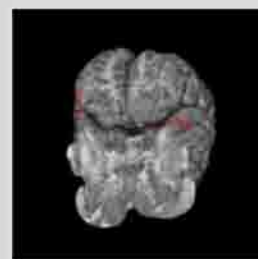
## **Apêndice C**

### **- Outros resultados**

Para o exemplo da grelha 1, 2, 3, 4 e 5 não foi possível usar a extracção automática. Apenas nos exemplos 6 e 7. Para os exemplos das grelhas 1, 2 a visualização do ECoG não é adequada, pois ao ser usado apenas uma fita de eléctrodos. E não é completamente eficaz para os exemplos das grelhas 3 e 5 pelo mesmo motivo.

No exemplo da grelha 6, são usadas 2 grelhas 4x8. Para este caso existem 3 exemplos particulares, a visualização da primeira grelha 4x8, da segunda grelha 4x8 e a visualização conjunta de ambas as grelhas, que por se encontrarem parcialmente sobrepostas, é possível a sua adaptação para uma visualização conjunta (ver Figura C.6).

# CASO 1

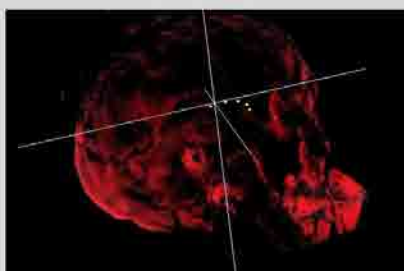


Esquema da grelha de eléctrodos:



2 fitas de eléctrodos (uma de cada lado)

Extracção:



Visualização:



Figura C.1: Caso 1

## CASO 2



Esquema da grelha de eléctrodos:



2 fitas de eléctrodos (uma de cada lado)

Extracção:



Visualização:

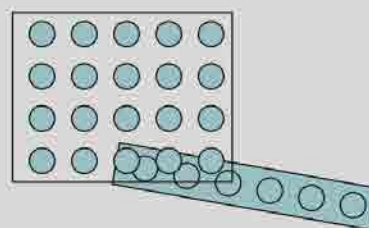
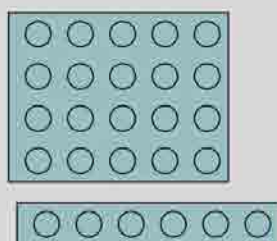


Figura C.2: Caso 2

## CASO 3

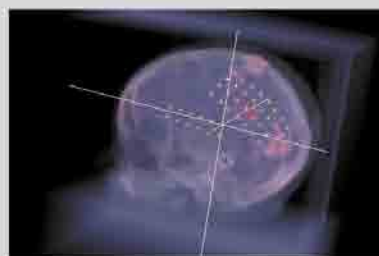
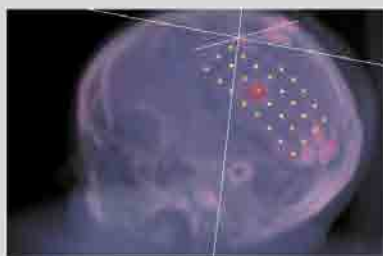


Esquema da grelha de eléctrodos:



posição relativa (ideia)

Extracção:



Visualização:

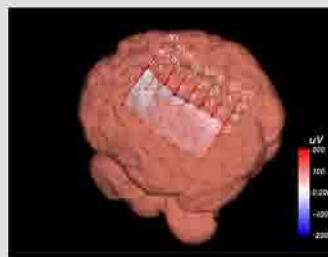
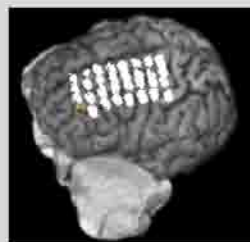


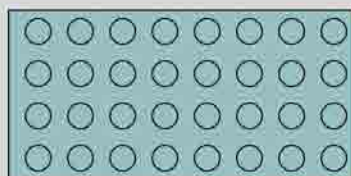
Figura C.3: Caso 3



## CASO 4



Esquema da grelha de eléctrodos:



Extracção:



Visualização:

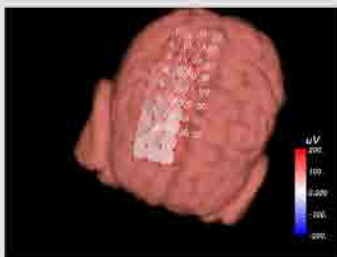
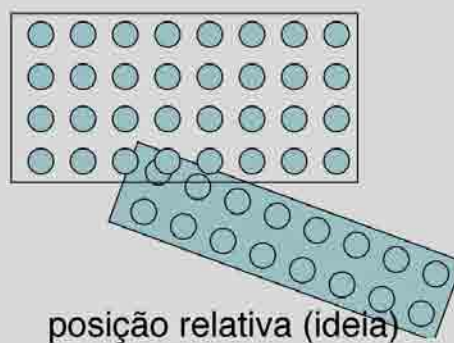
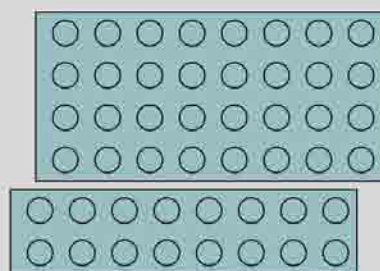


Figura C.4: Caso 4

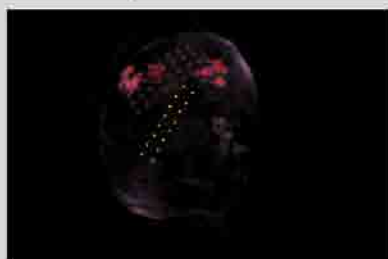
## CASO 5



Esquema da grelha de eléctrodos:



Extracção:



Visualização:

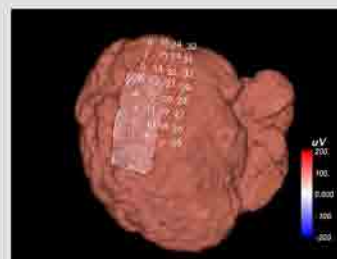
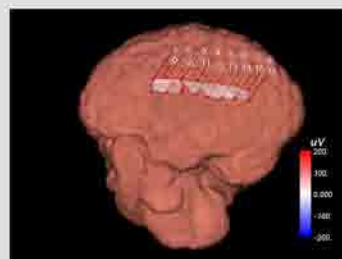


Figura C.5: Caso 5

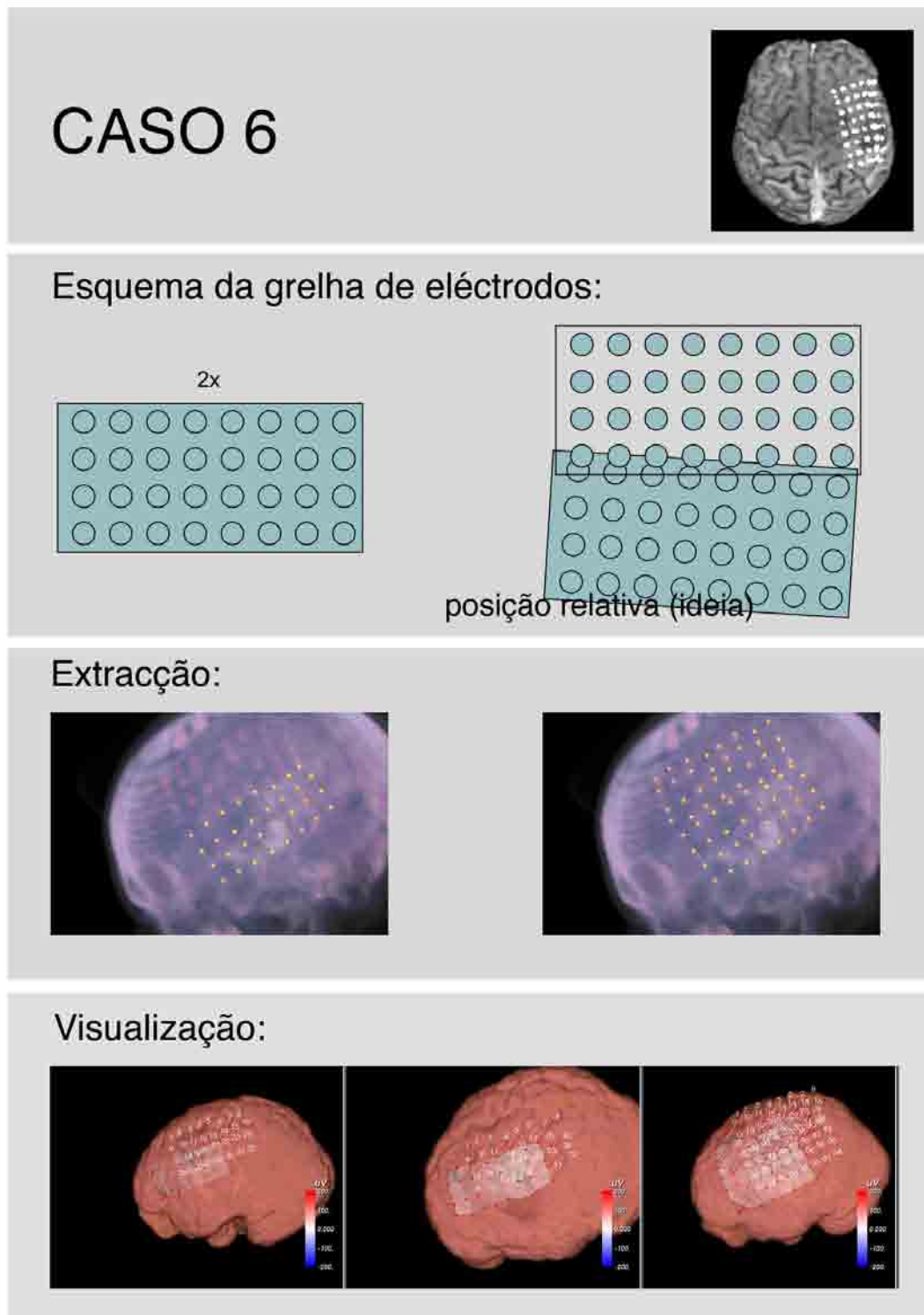


Figura C.6: Caso 6

# Apêndice D

## - Instalação

VTK - KWWIDGETS - WINDOWS HOW TO  
for Visual Studio 2005

You need the following packages:

Cmake:

cmake-2.4.6-win32-x86.exe

VTK source:

vtk-5.0.2.zip or vtk-5.0.2.tar.gz

KWWidgets source:

KWWidgetsNightlyCVS.tar.gz

TCL:

Active TCL

Run the Active TCL installer.

Run Cmake installer.

(NOTE: CMake installation folder and the projects it  
generates MUST be set to C:\)

Having TCL and Cmake ready. Unpack the VTK source and  
run Cmake indicating where the VTK source code folder  
and the destination folder.

->configure

then enable the flag VTK\_WRAP\_TCL.

->configure

->OK

You have the VTK directory ready for building.

Build and Install with MS2005.

Having VTK builded and installed. Unpack the KWWidgets source and run Cmake indicating where the KWWidgets source code folder and the destination folder.

->configure

->configure

->OK

You have the KWWidgets directory ready for building.

Build and Install with MS2005.

Now for compiling our own programs we will need to include the headers and libraries.

In MS2005

Tools->Options->Projects and Solutions->VC++ Directories and for Include files and Library files, include the necessary VTK, TCL and KWWidgets folders.

If you have installed correctly both VTK and KWWidgets most of them will be in:

Include files:

- c:\Program Files\VTK

- c:\Program Files\KWWidgets\include\KWWidgets

- c:\Tcl\include (depending where you installed TCL)

other folders might be necessary, add them if you need.

Library Files:

- c:\Program Files\VTK\lib

- c:\Program Files\KWWidgets\lib\KWWidgets

- c:\Program Files\KWWidgets\lib

- c:\Tcl\lib (depending where you installed TCL)

other folders might be necessary, add them if you need.